



UNIVERSIDAD DEL BÍO-BÍO
FACULTAD DE CIENCIAS EMPRESARIALES

Facultad de Ciencias Empresariales
Departamento de Sistemas de Información

“Sistema de ejercitación en línea para apoyar el aprendizaje de la programación en C”

Proyecto de Título presentado por Rodolfo Cuevas y Gerson Pereira de la carrera

Ingeniería Civil Informática

Dirigido por Sra. Brunny Troncoso

2021

Resumen

Este Proyecto se presenta para dar conformidad a los requisitos exigidos por la Universidad del Bío-Bío en el proceso de titulación para la carrera de Ingeniería Civil Informática. El proyecto se titula “Sistema de ejercitación en línea para apoyar el aprendizaje de la programación en C”.

La Universidad del Bío-Bío dicta dos carreras del área informática, las cuales poseen mallas distintas con asignaturas similares, dentro de estas asignaturas similares impartidas se encuentra el aprendizaje del lenguaje C, el cual genera una gran dificultad entre los estudiantes al momento de realizar ejercitación al ser su primer acercamiento a la programación. Debido a esto, y a la ausencia de herramientas que apoyen esta parte esencial en el aprendizaje se genera la necesidad de tener una herramienta capaz de facilitar el paso por esta asignatura.

Abstract

This project has been presented to satisfy the formalities required by Universidad del Bío-Bío in the Ingeniería Civil en Informática graduation process. The project is called “Exercise online system to support the learning C programming”

Universidad del Bío-Bío teaches two careers in the IT area, which have similar subjects, inside this similar subjects is the learning of C programming language, which has a great difficulty for the students at the moment of resolving exercises resulting from being their first approach to programming. Because of this, and the absence of tools capable of facilitate the passage through this subject, the need of having a tool capable of support the passage through this subject is generated.

Índice General

1	Introducción	11
2	Definición de la empresa o institución	12
2.1	Descripción de la empresa	12
2.2	Descripción del área de estudio	15
2.3	Descripción de la problemática	17
2.4	Estado del Arte	19
2.4.1	Proyecto: Sistemas Inteligentes Aplicados a la Enseñanza de la Programación en Ingeniería	19
2.4.2	Ambiente Integrado de Visualización de Estructuras de Datos (VIA-ED)	20
2.4.3	UVaOnlineJudge [uHunt]	20
2.4.4	Light OJ	21
2.4.5	Sphere online judge	22
3	Definición proyecto	24
3.1	Objetivos del proyecto	24
3.1.1	Objetivo General	24
3.1.2	Objetivos Específicos	24
3.2	Ambiente de Ingeniería de Software	25
3.3	Definiciones, Siglas y Abreviaciones	26
4	Especificación de requerimientos de Software	27
4.1	Alcances	27
4.2	Objetivo del software	27
4.2.1	Objetivos Generales	27
4.2.2	Objetivos Específicos	28
4.3	Descripción Global del Producto	28
4.3.1	Interfaz de usuario	28
4.3.2	Interfaz De Hardware	29
4.3.3	Interfaz Software	29
4.3.4	Interfaces de comunicación	29
4.4	Requerimientos Específicos	30
4.4.1	Requerimientos Funcionales del sistema	30
4.4.2	Interfaces externas de entrada	32
4.4.3	Interfaces externas de Salida	32
4.4.4	Atributos del producto	33
5	Factibilidad	34
5.1	Factibilidad técnica.	34

5.2 Factibilidad operativa.	35
5.3 Factibilidad económica.	36
5.4 Conclusión de la factibilidad	38
6 Análisis	39
6.1 Procesos de Negocios futuros	39
6.2 Diagrama de Flujo de Datos	40
6.3 Diagrama de casos de uso	41
6.3.1 Actores	41
6.3.2 Casos de Uso y descripción	41
6.3.3 Especificación de los Casos de Uso	43
6.3.3.1 Caso de Uso: <Autenticar Usuario>	43
6.3.3.2 Caso de Uso: <Ingresar Problemas>	44
6.3.3.3 Caso de Uso: <Consultar Problema>	45
6.3.3.4 Caso de Uso: <Modificar Problema>	46
6.3.3.5 Caso de Uso: <Eliminar Problema>	47
6.3.3.6 Caso de Uso: <Crear Cuaderno>	48
6.3.3.7 Caso de Uso: <Consultar Cuaderno>	49
6.3.3.8 Caso de Uso: <Modificar Cuaderno>	50
6.3.3.9 Caso de Uso: <Eliminar Cuaderno>	51
6.3.3.10 Caso de Uso: <Crear Usuario>	52
6.3.3.11 Caso de Uso: <Modificar Usuario>	53
6.3.3.12 Caso de Uso: <Eliminar Usuario>	54
6.3.3.13 Caso de Uso: <Consultar Rendimiento>	55
6.3.3.14 Caso de Uso: <Ver Cuaderno>	56
6.3.3.15 Caso de Uso: <Resolver problema>	57
6.3.3.16 Caso de Uso: <Procesar Respuesta del Alumno>	58
6.3.3.17 Caso de Uso: <Evaluar Solución>	59
6.3.3.18 Caso de Uso: <Consultar Problemas Resueltos>	60
6.3.3.19 Caso de Uso: <Consultar Curso>	61
6.3.3.20 Caso de Uso: <Crear Curso>	62
6.3.3.21 Caso de Uso: <Eliminar Curso>	63
6.3.3.22 Caso de Uso: <Inscribir Alumno>	64
6.3.3.23 Caso de Uso: <Desinscribir Alumno>	65
6.3.3.24 Caso de Uso: <Enviar Problema Simple>	66
6.3.3.25 Caso de Uso: <Enviar Problema Aleatorio>	67
6.3.3.26 Caso de Uso: <Crear Carrera>	68
6.3.3.27 Caso de Uso: <Eliminar Carrera>	69
6.4 Modelamiento de datos	70
7 Diseño	71

7.1	Diseño Físico de la Base de datos	71
7.2	Diseño de arquitectura funcional	72
7.3	Diseño interfaz y navegación	73
7.4	Especificación de módulos	77
8	Pruebas	80
8.1	Elementos de prueba	80
8.2	Especificación de las pruebas	81
8.3	Responsables de las pruebas	83
8.4	Calendario de pruebas	84
8.5	Detalle de las pruebas	84
8.6	Conclusiones de Prueba	84
9	Conclusiones	85
9.1	Trabajos futuros	86
10	Bibliografía	87
11	Anexo: Planificación Inicial del Proyecto	88
12	Anexo: Resultados de Iteraciones en el Desarrollo	91
13	ANEXO: Manual del Usuario	97
13.1	Administrador	97
13.2	Profesor	110
13.3	Alumno	126
14	Anexo: Especificación de las Pruebas	134
14.1	Pruebas de Unidad	134
14.1.1	<Ingreso de Problemas>	134
14.1.2	<Ingreso de casos de Pruebas>	134
14.1.3	<Ingreso de Respuesta de Alumno>	135
14.1.3	<Ingreso de Datos Alumnos>	136
14.1.4	<Ingreso de Datos Profesor>	136
14.1.5	<Ingreso de Curso>	137
14.1.6	<Ingreso de Carrera>	138
15	Anexo: Diccionario de Datos del Modelo de Datos	139

Índice de Tablas

Tabla 1: Requerimientos Funcionales	30
Tabla 2: interfaces externas de entrada	32
Tabla 3: Interfaces externas de salida	32
Tabla 4: Gastos estimados proyecto	36
Tabla 5: Gastos de electricidad y conectividad	37
Tabla 6: Especificación de las pruebas	81
Tabla 7: Especificación de prueba ingresar problema	134
Tabla 8: Especificación prueba ingreso de casos de prueba	134
Tabla 9: Especificación prueba ingreso de respuesta alumno	135
Tabla 10: Especificación prueba ingreso de datos de alumno	136
Tabla 11: Especificación prueba ingreso de datos de profesor	136
Tabla 12: Especificación prueba ingreso de datos de curso	137
Tabla 13: Especificación prueba ingreso de datos de carrera	138

Índice de Figuras

Figura 2.1.1: Organigrama Universidad del Bío-Bío	14
Figura 2.2.1: Estructura Organizacional Facultad de Ciencias Empresariales	16
Figura 2.3.1: Estructura general de una asignatura	18
Figura 6.1.1: Diagrama de BPMN	39
Figura 6.2.1: Diagrama de flujo de datos nivel de contexto	40
Figura 6.2.2: Diagrama de flujo de datos de nivel superior	40
Figura 6.3.2.1: Casos de uso general, actores principales	42
Figura 6.3.2.2: Casos de uso del sistema web	43
Figura 6.4.1: Modelo Entidad Relación	70
Figura 7.1.1: Diagrama modelo físico de la base de datos	71
Figura 7.2.1: Árbol de descomposición del sistema	72
Figura 7.3.1: Diseño de interfaz general	73
Figura 7.3.2 : Jerarquía de menú	74
Figura 7.3.3: Esquema de navegación administrador	75
Figura 7.3.4: Esquema de navegación profesor	76
Figura 7.3.5: Esquema de navegación alumno	76
Figura 12.1: Base de Datos MYSQL	91
Figura 12.2: Inicio de sesión	92
Figura 12.3: Menú de Opciones Profesor	92
Figura 12.4: Menú Opciones Alumno	93
Figura 12.5: Mis Cursos Alumno	94
Figura 12.6: Subida Respuesta Alumno	95
Figura 12.7: Menú de Opciones Administrador	96
Figura 12.8: Módulo Importar Curso Archivo xlsx	96
Figura 12.9: Espacio para Importar Curso Perfil Profesor	96
Figura 13.1.1: Menú de Administrador	97
Figura 13.1.2: Opciones del menú Cursos(Todos)	98
Figura 13.1.3: Crear Curso Administrador	99
Figura 13.1.4: Importar Curso Administrador	100
Figura 13.1.5: Crear Curso Importado Administrador	100

Figura 13.1.6: Completar Importación Administrador	101
Figura 13.1.7: Participantes de un curso Administrador	102
Figura 13.1.8: Eliminar un Alumno Administrador	102
Figura 13.1.9: Cuaderno Alumno parte 1 Administrador	103
Figura 13.1.10: Cuaderno Alumno parte 2 Administrador	103
Figura 13.1.11: Menú Problemas Administrador	104
Figura 13.1.12: Mostrar Respuesta Problema Administrador	105
Figura 13.1.13: Agregar Problema Administrador	106
Figura 13.1.14: Procesar Pregunta Administrador	106
Figura 13.1.15: Confirmar Respuesta Administrador	107
Figura 13.1.16: Menú Profesores Administrador	107
Figura 13.1.17: Agregar Profesor Administrador	108
Figura 13.1.18: Menú Carrera Administrador	109
Figura 13.1.19: Agregar Carrera Administrador	109
Figura 13.2.1 Opciones Menú Profesor	110
Figura 13.2.2 Opciones de Cursos	111
Figura 13.2.3 Crear Curso Profesor	111
Figura 13.2.4 Importar Curso Profesor	112
Figura 13.2.5 Crear Curso Importado Profesor	113
Figura 13.2.6 Completar Importación	113
Figura 13.2.7: Participantes de un Curso	114
Figura 13.2.8: Eliminar un Alumno del Curso	114
Figura 13.2.9: Cuaderno Pendiente Alumno	115
Figura 13.2.10: Cuaderno Resuelto Alumno	115
Figura 13.2.11: Menú de Problemas Profesor	116
Figura 13.2.12: Mostrar Respuesta del Profesor	117
Figura 13.2.13: Agregar Problema Profesor	118
Figura 13.2.14: Procesar Pregunta Profesor	118
Figura 13.2.15: Confirmar Problema y Respuesta Profesor	119
Figura 13.2.16: Envío de Problemas Profesor	120
Figura 13.2.17: Enviar Problema Individual Profesor	121
Figura 13.2.18: Seleccionar Problema a Enviar Profesor	122

Figura 13.2.19: Enviar Problema Individual	122
Figura 13.2.20: Enviar Problema de Forma Aleatoria Profesor	123
Figura 13.2.21: Menú Alumnos Profesor	124
Figura 13.2.22: Agregar Nuevo Alumno	125
Figura 13.2.23: Perfil de Profesor	126
Figura 13.3.1: Menú de Alumno	126
Figura 13.3.2: Dashboard de Alumno	127
Figura 13.3.3: Mis Cursos de Alumno	128
Figura 13.3.4: Problemas Cuaderno Alumno	129
Figura 13.3.5: Solución Errónea Alumno	130
Figura 13.3.6: Solución Correcta Alumno	131
Figura 13.3.7: Mis Problemas Alumno	132
Figura 13.3.8: Ingreso Respuesta Alumno	133

1 INTRODUCCIÓN

Actualmente, las herramientas informáticas se han convertido en un apoyo docente en todas las universidades y, lo que es más importante, un acceso a Internet que ayuda en el estudio e investigación de estudiantes y profesores.

A día de hoy, La Universidad del Bío-Bío tiene múltiples plataformas para apoyar la docencia para promover el desarrollo de materiales de la asignatura y la comunicación profesor-estudiantes. Las plataformas existentes de la universidad son Moodle y Adecca específicamente para los estudiantes de la universidad. Se calcula que estas plataformas cuentan con varios temas de programación, por lo que requieren un soporte básico y fundamental para la comprensión y el aprendizaje de los estudiantes.

El proyecto consiste en crear un sistema web que brinde apoyo a la docencia de las distintas asignaturas ofertadas por la universidad para la ejercitación de los alumnos y detectar los puntos más débiles dentro de la asignatura y los temas planteados, para de esta forma lograr menos deserción dentro de las carreras en que se imparten asignaturas ligadas a la programación en C.

Lo siguiente a tratar es la base teórica para sustentar el proyecto a desarrollar. Para ello, se realizó una investigación y consultas con el propósito de comprender las herramientas que se pueden utilizar para construir el sistema.

El sistema se describe en diferentes capítulos cada uno por fases como Requisitos y especificaciones de desarrollo de software. Los requerimientos de usuario en detalle, todos los aspectos relevantes del sistema, diferentes tipos de viabilidad en el proyecto, análisis de problemas y diseño de soluciones, posteriores pruebas del sistema y el desarrollo del software en todas sus etapas.

2 DEFINICIÓN DE LA EMPRESA O INSTITUCIÓN

2.1 Descripción de la empresa

Antecedentes generales de la Empresa

- Nombre: Universidad del Bío-Bío
- Dirección: Avenida Collao 1202
- Rubro: Universidades
- La Universidad del Bío-Bío se dedica a contribuir mediante el cultivo del saber, de la educación superior, de la investigación, de la asistencia técnica y de la capacitación, a la formación de profesionales y al desarrollo regional en el territorio en el cual realiza sus actividades, sin perjuicio de poder extender sus actividades, si las condiciones así lo requieren al ámbito nacional e internacional.
- Competencia directa.

En la octava región se encuentran cuatro universidades del consejo de rectores, todas ubicadas en la ciudad de Concepción o sus alrededores. Por lo que la competencia directa de la Universidad del Bío-Bío, son la Universidad de Concepción, Universidad Católica de la Santísima Concepción y la Universidad Técnica Federico Santa María, además se encuentran algunas universidades privadas como la Universidad San Sebastián, Universidad de las Américas, Santo Tomás, INACAP y así mismo algunos institutos profesionales como: AIEP, IPP y Virginio Gómez, todos ellos cuentan con carreras relacionadas al área Informática, por lo que se consideran también como competencia para la institución.

- Cuota de mercado.

La universidad del Bío-Bío cuenta con 13259 estudiantes de un total de 159510 registrados en universidades, por lo que la cuota de mercado que cubre la institución es de un 8,31% de la cuota de mercado.

- Misión y Visión:

Misión: Crear y transmitir el conocimiento mediante la docencia, la investigación, la innovación, la creación artística y la interacción con el entorno social en las diversas áreas del conocimiento y la cultura, basada en la responsabilidad social y en los principios de excelencia, pluralismo, transparencia y equidad de género (Universidad del Bío-Bío, n.d.).

Visión: Ser una Universidad comprometida con su carácter estatal y birregional, innovadora, inclusiva y compleja, con proyección nacional e internacional, que contribuye al progreso sostenible de su territorio y sus habitantes, de excelencia en la formación de personas y con sentido de equidad social y de género (Universidad del Bío-Bío, n.d.).

Estructura Organizativa.

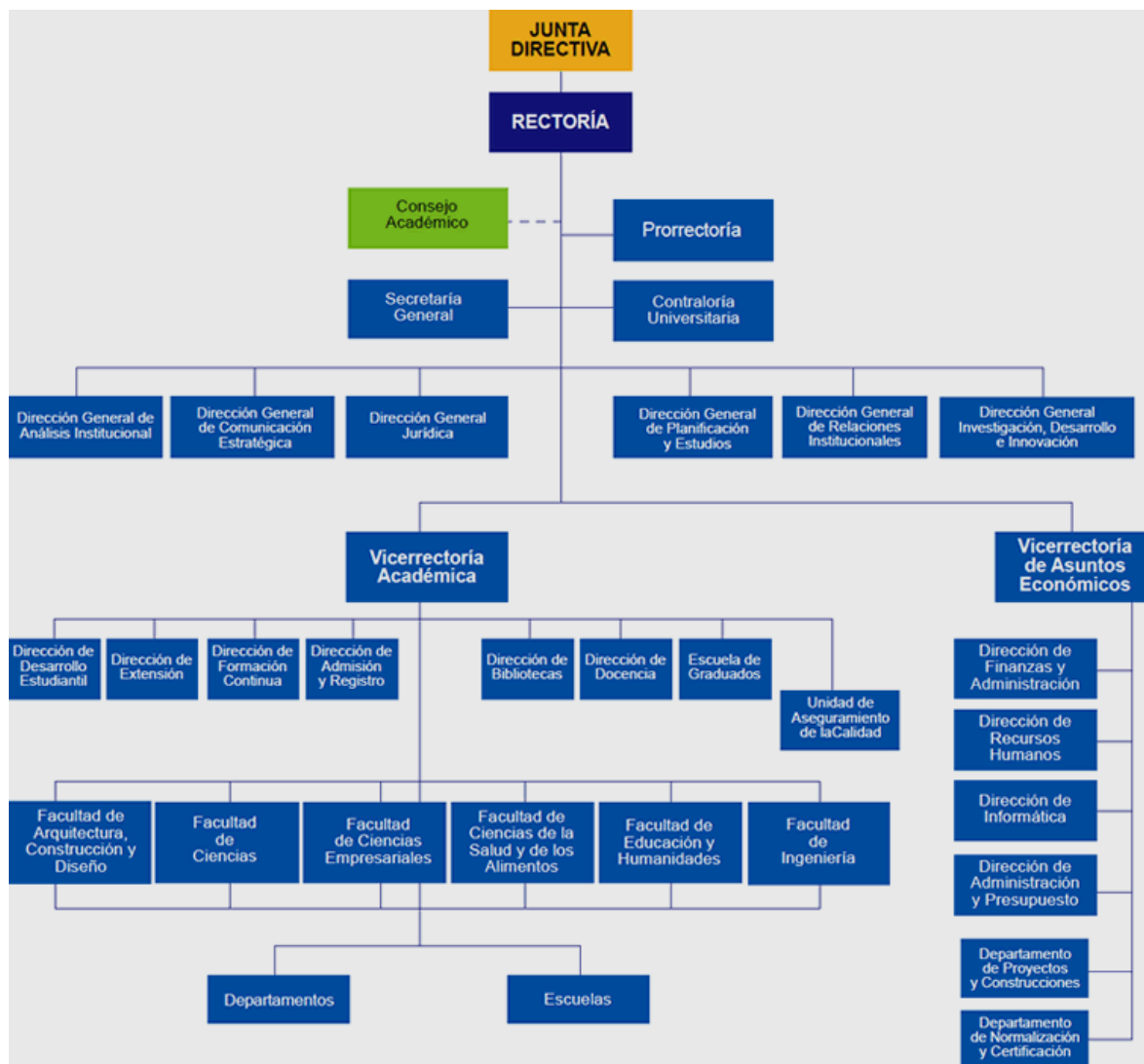


Figura 2.1.1: Organigrama Universidad del Bío-Bío

Como se aprecia en la figura , toda la estructura organizativa que rige a la universidad desde los mandos más altos hasta los departamentos y escuelas de la Universidad se encuentra regida y estructurada desde la Junta directiva que es la que toma las decisiones, que afectan a cada dirección , facultad, departamentos, escuelas y carreras.

2.2 Descripción del área de estudio

El sistema a desarrollar durante esta tesis está dirigido a la universidad del Bío-Bío y también para la facultad de Ciencias Empresariales y el departamento de sistemas de información el cual se relaciona con las carreras de Ingeniería Civil Informática (ICI) e Ingeniería Ejecución en Computación e Informática (IECI) contando cada carrera con su respectiva escuela y con sus respectivos jefes/as de carrera teniendo relación directa con el decano de la facultad, los cuales están encargados junto con los directivos de cada carrera de ver que asignaturas se dictan en un semestre y cuales son los profesores que realizan estas asignaturas. Como el sistema está propuesto y pensado para las asignaturas que trabajen en totalidad, en módulos, programas y/o códigos para el desarrollo y profundización en lenguaje C que está directamente relacionado con los profesores que impartirán estas asignaturas en el futuro cercano.

- Objetivos del Departamento de Sistemas de Información
 - ❖ El desarrollo académico de la Ciencias de la Computación e Informática y la participación activa en la formación, capacitación y asistencia técnica de profesionales en informática, particularmente en las áreas de la informática aplicada a la gestión.
 - ❖ Una creciente actividad de investigación relevante, buscando la formación de equipos en áreas donde se pueda destacar.
 - ❖ Un permanente perfeccionamiento de sus cuadros académicos y capacitación del personal administrativo, para el mejoramiento de la calidad.
 - ❖ Establecimiento de relaciones con el entorno, que permita una mayor pertinencia y contribución a su quehacer.

❖ Estructura organizativa del área de estudio

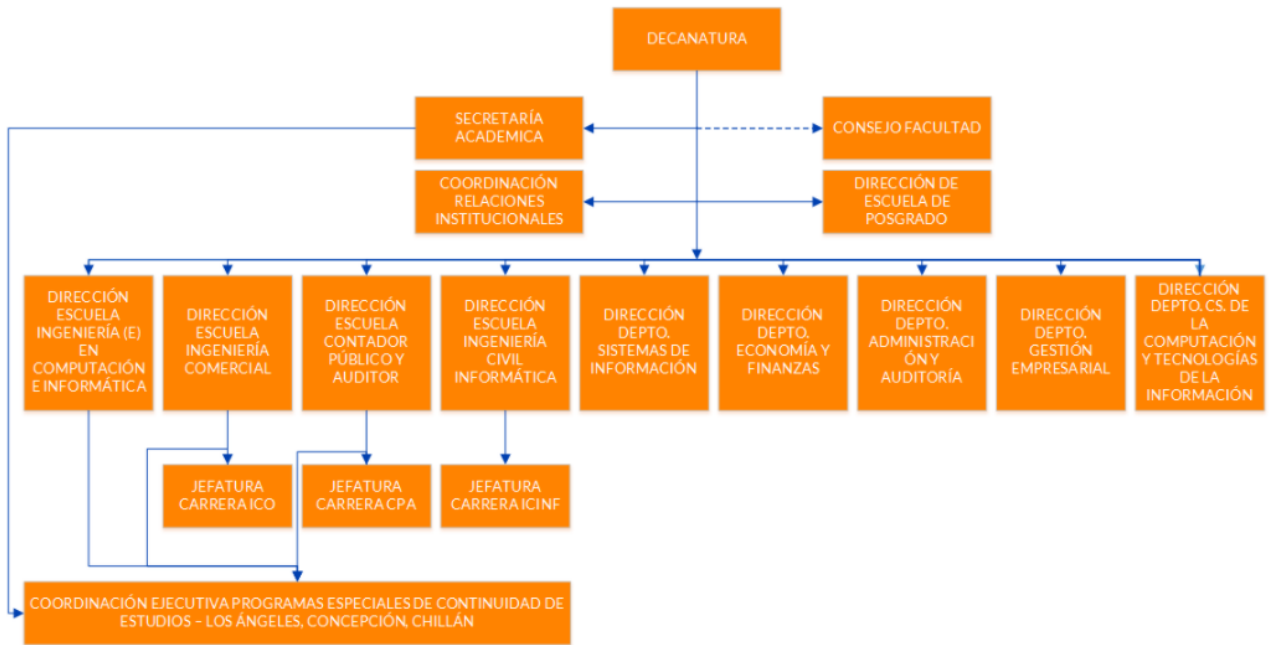


Figura 2.2.1: Estructura Organizacional Facultad de Ciencias Empresariales.

La Dirección del DSI según lo señalado en el organigrama de la facultad es responsable directamente ante el Decano de la facultad, trabaja en conjunto con las demás escuelas y departamentos dentro de la misma con el fin de organizar la docencia y la investigación en diferentes áreas afectadas por su participación. También, se responsabiliza de todas las comunicaciones oficiales con el decano, otros funcionarios de la universidad y las comunicaciones con los estudiantes de la facultad y asimismo para los demás estudiantes en la universidad.

2.3 Descripción de la problemática

Las asignaturas en las que se aplica desarrollo de programación en lenguaje C son variadas a lo largo de la carrera partiendo desde el primer semestre tanto para ICI como para IEI, los alumnos que cursan las asignatura más avanzadas a partir el segundo y tercer semestre conocen las bases de la programación (recorridos, búsquedas, ordenación, acciones/funciones, paso de parámetros, etc.) pero en general gran parte de ellos o la totalidad tienen problemas con algunas de estas bases, lo que ocasiona dificultades al momento de aprender métodos o funciones más avanzadas.

El trabajo personal del alumno y la resolución de problemas es fundamental para el aprendizaje de los conceptos básicos de programación. El contenido de los cursos de programación es acumulativo, cada nuevo tema se basa en lo desarrollado en temas anteriores, al tratarse de conceptos prácticos, la resolución de problemas y la implementación de programas que aplican los aspectos desarrollados en las sesiones teóricas, son elementos esenciales para aprender la materia. Pero desde la experiencia docente se percibe un fracaso del sistema tradicional de enseñanza en la programación porque una cantidad significativa de los alumnos no hacen las tareas o lo hacen parcialmente y al momento de realizarlas no saben si es la solución correcta o la óptima para el problema, lo que genera incertidumbre y retrasos en su aprendizaje, situación que los profesores no pueden detectar a tiempo en gran parte de los cursos, además la cantidad de alumnos en los diferentes cursos hace difícil el monitoreo de cada actividad y ayudar a cada uno en forma individual es aún más complejo.

El funcionamiento de las asignaturas en la actualidad es de la siguiente manera:

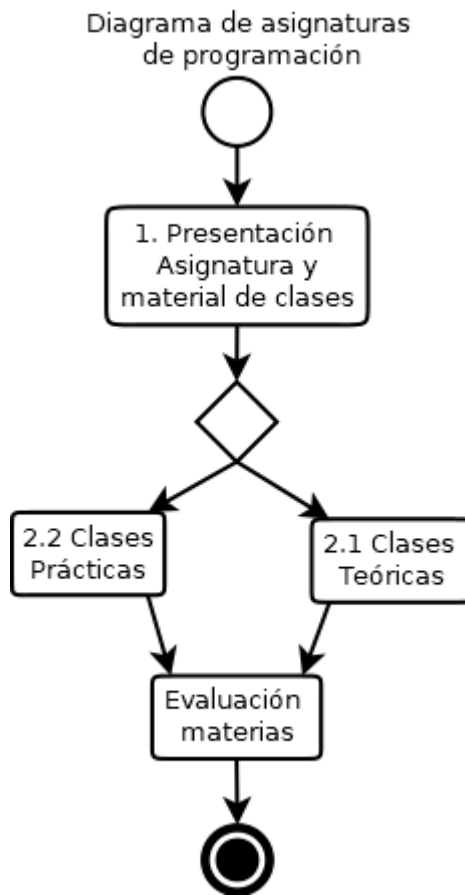


Figura 2.3.1: Estructura general de una asignatura.

Por lo general las asignaturas se distribuyen de esta manera, siempre se dividen en clases teóricas y prácticas abarcando las teóricas hasta el 60% o más de las horas del semestre, por lo que en gran parte de los casos no se alcanza a ver con profundidad todo el manejo y funciones de los algoritmos concluyendo siempre en evaluaciones en la parte práctica que no termina de medir todo el conocimiento deseado.

2.4 Estado del Arte

Para satisfacer la problemática planteada se ha investigado sobre las diferentes herramientas con las cuales se podría desarrollar un sistema de ambiente educativo virtual. Las webs que contienen jueces evaluadores de problemas no son tan raras como podríamos llegar a pensar, por ello es importante ver algunos de ellos centrándonos en aspectos importantes, cómo sus características y funcionamiento con él fin de qué sirvan como puntos de referencia para desarrollar nuestra propia plataforma haciendo distinciones en puntos claves.

2.4.1 Proyecto: Sistemas Inteligentes Aplicados a la Enseñanza de la Programación en Ingeniería

El proyecto objeto de esta presentación, se desarrolla actualmente en la Facultad de Ciencias Exactas, Físicas y Naturales de la Universidad Nacional de Córdoba. Ha sido presentado para su acreditación ante la Secretaría de Investigación y en Ciencia y Tecnología de esta Universidad en la convocatoria bianual 2012-2013 de proyectos I+D. Es el primer antecedente formal en esta línea de investigación aplicada a la enseñanza en Ingeniería en esta Casa de Estudios.

Su mayor relevancia consiste en el impacto que se espera de él para las carreras de Ingeniería, especialmente la de Computación, tanto para el rendimiento académico de los estudiantes como para el fortalecimiento de los recursos humanos-investigadores, docentes y estudiantes-en investigación.

El eje central consiste en la utilización de sistemas tutores inteligentes y aplicaciones con robots para la enseñanza de Programación en Ingeniería.

Algunos métodos utilizados para este proyecto son:

- Determinación de modelos mentales de aprendizaje, entre los estudiantes de cursos de programación de Ingeniería, a partir de información disponible de las cursadas 2010-2011, usando redes neuronales como herramienta para detectar patrones.

- Elaboración de la ontología de base para la enseñanza de la programación y producción de las reglas de restricciones para construir el mapa conceptual de la asignatura.
- Puesta a punto y prueba del STI con estudiantes de 2013, validando los resultados obtenidos mediante la comparación con STI basado en teorías de prueba.
- Desarrollo e implementación de actividades prácticas utilizando Python y C++, nativos para RUR-PLE y compatibles con LEGO. Aplicación de las actividades.

2.4.2 Ambiente Integrado de Visualización de Estructuras de Datos (VIA-ED)

Este modelo tiene presente la caracterización de las Técnicas de Visualización y las deficiencias encontradas en el diseño e implementación eficiente de estructuras de datos, se propone un ambiente basado en mapas conceptuales, que organiza e integra el modelo del conocimiento de esta materia y sirve de interfaz visual y repositorio de recursos informáticos (un Aula Virtual, simulaciones, animaciones, imágenes y documentos en diferentes formatos), útiles para el autoaprendizaje en un modelo de estudios semipresencial. Incluye, el Sistema de Visualización dinámica de Programas VisualProg, como su principal recurso de aprendizaje. VIA-ED permite:

- Facilitar la visualización dinámica de los datos y del código de programas
- Mostrar las operaciones
- Permitir la realización de cambios en el código del programa y comprobar el efecto que provoca en los datos.

2.4.3 UVaOnlineJudge [uHunt]

Este es el juez de la universidad de Valladolid y puede ser usado como guía para el proyecto. Su sencillez hace que sea bastante intuitivo de usar y por eso hemos decidido adaptar ciertos detalles a nuestra interfaz.

Sólo es necesario estar registrado para tener acceso a las opciones de:

- Enviar un problema, para ello sólo será necesario seleccionar el tipo de lenguaje, el id del problema y el código del mismo, que podrá escribirse en un cuadro de texto o subirse directamente desde un fichero de tu ordenador.
- Ver/modificar tu perfil, en este apartado se muestran los datos que proporcionamos al crear la cuenta, y se da la opción de cambiarlos. También se da la posibilidad de subir una foto para mostrarla en el perfil.
- Ver tus estadísticas. Proporciona datos del usuario tales como estadísticas con gráficos en forma de tarta, número de envíos totales, número problemas intentados y resueltos, fecha del primer y del último envío y una tabla con todos los problemas que has resuelto.
- Ver tus últimos envíos, en esta página se muestran los últimos envíos que el usuario ha realizado, proporcionando además del ID del envío, el ID del problema, el título, el veredicto, el lenguaje, el tiempo de ejecución y la fecha de envío.
- Los problemas se clasifican por categorías, según sean problemas normales o de concurso. Estos últimos se agrupan además por el concurso en el que se plantearon. Cada una de estas categorías puede estar organizada por volúmenes (compuestos por 100 problemas), por autor, por año o por título.

2.4.4 Light OJ

Este juez evaluador de problemas fue creado por Jane AlamJam. El lenguaje que se usa en esta página es el inglés, y no dispone de ninguna opción para cambiarlo. Se necesita estar registrado para hacer uso de su sistema.

Opciones del menú:

- Home: muestra la página de inicio, con un mensaje de bienvenida, también puedes ver tu perfil.

- Volume: permite ver los problemas (organizados por volúmenes), categorías, enviar un problema o ver tus envíos.
- Contest: permite ver los concursos que están abiertos en ese momento además de los concursos para practicar.
- Training: para entrenar las habilidades.
- Community: permite ver los foros.
- Credits: muestra quien lo ha desarrollado y unos agradecimientos.

Los problemas están agrupados, por defecto, en volúmenes de 100 problemas cada uno pudiendo verlos además agrupados por categoría o por autor. Lo más destacable de esta última agrupación es que si seleccionamos un autor nos muestra una lista con todos los problemas que ha creado.

2.4.5 Sphere online judge

Este juez es bastante amigable, el idioma que usa es el inglés y no dispone de ninguna opción para cambiarlo por otro. Es destacable la cantidad de publicidad que hay en todas sus páginas, y además no está relacionada con el tema de los algoritmos, para poder resolver y enviar problemas es necesario estar registrado. Consta de un menú lateral izquierdo fijo en el que se muestran todas las opciones de las que dispone la web, según la opción seleccionada cambia el contenido de la página.

Consta de lo siguiente:

- Register: enlaza al formulario de registro.

- Tutorials: esta página muestra enlaces a cuatro tutoriales sobre nuevos usuarios, problemas, problemas interactivos, problemas de concurso. También hay información de que es spoj y una pequeña guía de cómo empezar.
- Status: muestra una tabla con los últimos envíos.
- Submit: enlace a la página de enviar un problema
- Problems: muestra la lista de problemas.
- Search: es un cuadro de búsqueda de google.
- news: es la página principal, muestra información de concursos, número de total de envíos, número de usuarios registrados, número de problemas publicados, total de problemas, número de instituciones afiliadas, número de concursos afiliados, número de lenguajes de programación
- Foro: tiene acceso a un foro, pero es independiente de la página

La clasificación de los problemas se puede hacer de cinco formas, por el id del problema, por reto, por problemas para aficionados, problemas sencillos, lista de problemas de enigma.

Una vez analizados todos estos proyectos y aplicaciones independientes se tomarán como referencia y guía para la creación del sistema web y sus módulos que se trabajarán y desarrollarán a lo largo de este proyecto.

3 DEFINICIÓN PROYECTO

3.1 Objetivos del proyecto

3.1.1 Objetivo General

- Apoyar, a través del uso de TI, el desarrollo de competencias de programación en C en los alumnos de algunas asignaturas ofertadas por el Departamento de Sistemas de Información de la Facultad de Ciencias Empresariales de la Universidad del Bío-Bío.

3.1.2 Objetivos Específicos

- Desarrollar un estudio del arte del desarrollo de plataformas educativas similares, sus aplicaciones y características técnicas con el fin de seleccionar las herramientas de desarrollo y capturar ideas de utilidad para el desarrollo del proyecto.
- Desarrollar un repositorio de distintos problemas base, que facilite a los profesores de las asignaturas de programación ingresar y compartir nuevos problemas, catalogados según los temas tratados y el nivel de dificultad de cada uno de ellos. Se espera cubrir el rango de dificultad desde los ejemplos más básicos hasta programación de árboles binarios, binarios de búsqueda, heap y otros.
- Diseñar un sistema que gestione la generación de distintos cuadernos de problemas para los estudiantes de cada asignatura, facilitando la corrección automática de los problemas de programación presentados y generando reportes de su desempeño académico, tanto de manera individual como colectiva.
- Implementar y probar la solución propuesta.

3.2 Ambiente de Ingeniería de Software

- Metodología de desarrollo.

La metodología a utilizar a lo largo de este proyecto será evolutiva debido a que facilita un desarrollo de forma creciente, se logra un mayor entendimiento del sistema y ofrece una mejor retroalimentación la cual ayuda a la inexperiencia de los alumnos tesistas con el tipo de sistema que se trabajará, ofreciendo algunas ventajas:

- “En la metodología evolutiva se asume que los requisitos pueden cambiar en cualquier momento del ciclo de vida y no solo en la etapa de análisis, sino siendo variable en toda etapa” (Programación Estructurada, n.d.).

- La ventaja de este modelo es que se puede obtener una buena retroalimentación de los usuarios durante cada iteración de las actividades de especificación, desarrollo y prueba.

- Técnicas y notaciones.

Las técnicas que se utilizaran en el desarrollo de software son la utilización de la plantilla para la creación del documento, la técnica de Casos de uso, reuniones con la profesora Sra. Brunny Troncoso como profesora guía, el modelo entidad-relación, diagrama para representar la jerarquización del menú, esquemas de navegación y lluvia de ideas para recopilar los requisitos, y abarcar las necesidades principales del software durante las reuniones.

La notación que se utiliza en casos de usos es UML 1.0, en el modelo entidad-relación se encuentra la notación MPM1999, además de una adaptación basada en IEEE Software requirements Specifications Std 830-1998.

- Estándares de documentación, producto o proceso.

ISO/IEC 9126: Para identificación de requerimientos no funcionales. Adaptación basada en IEEE Software requirements Specifications Std 830-1998: Para la especificación de requerimientos de software.

- Herramientas de apoyo al desarrollo de software que serán utilizadas.

Enterprise Architect: Para diagramar procesos (BPMN), Casos de usos.

Power Designer: Para diagramar la base de datos, además de MER y MR.

Editores de texto: Como Visual Studio Code, para crear, modificar e implementar código de programación.

Navegadores: Como Google Chrome, para probar e ir analizando código ya escrito.

Balsamiq Mockups Portable: Para el diseño de las diferentes pantallas del software.

LucidChart: Para el diseño del modelo de casos de uso(colaborativamente).

Photoshop: Para diseño de imágenes usadas en el sitio web.

3.3 Definiciones, Siglas y Abreviaciones

- ICI: Ingeniería Civil en Informática.
- IECI: Ingeniería Ejecución en Computación e Informática.
- URL: Secuencia de caracteres que hace referencia a sitios y recursos de internet , es su identificador.
- PHP: Lenguaje de programación para el desarrollo de páginas web.
- JS: Lenguaje de programación conocido cómo Java Script, para distintos usos cómo páginas web y aplicaciones web.
- MySQL: Gestor de bases de datos relacionales para el almacenamiento de información importante y de identificación.
- VSC: Visual Studio Code, editor de texto sofisticado para código de programación en múltiples lenguajes.
- HTML: Lenguaje de marcado que se utiliza para el desarrollo de páginas de Internet.
- IT: Information technologies (Referencia en inglés a tecnologías de la información).
- TI: tecnología de la información.

4 ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE

4.1 Alcances

Al ser un recurso exclusivo del lenguaje de programación en C, el software tiene las siguientes particularidades:

- Mantener un repositorio online con distintos problemas que permitan la creación de un cuaderno de ejercicios con diferentes problemas para cada alumno.
- Brindar una corrección automática de los problemas, señalando a cada estudiante sus errores de haberlos cometido, junto con guardar las respuestas en un “hoja de respuestas”.
- Permitir la generación de informes de desempeño individual y grupal para que los profesores puedan consultarlo si lo necesitan.
- Permitir que los profesores puedan agregar preguntas al repositorio de preguntas existentes
- Mantener una jerarquización dentro del software, para mantener la seguridad e integridad del sistema.

4.2 Objetivo del software

4.2.1 Objetivos Generales

- Desarrollar un repositorio de distintos problemas base, que facilite a los profesores de la asignatura ingresar y compartir nuevos problemas, catalogados según los temas tratados y el nivel de dificultad de cada uno de ellos. Se espera cubrir el rango de dificultad desde los ejemplos más básicos hasta programación de árboles binarios, binarios de búsqueda, heap y otros.

- Dar soporte a la corrección automática de los problemas. Para ello, cada problema base debe llevar asociado las pautas que hagan posible su corrección automática (varios test de pruebas con entradas y salidas esperadas para cada una de ellas).
- Gestionar la generación de distintos cuadernos de problemas para los estudiantes. Cada cuaderno está organizado por temas en los que se incorporan diferentes problemas según los tópicos de cada asignatura.

4.2.2 Objetivos Específicos

- Informar a cada estudiante de la evaluación (automática) de sus soluciones, brindándole la oportunidad de volver a ingresar una respuesta hasta obtener una que sea correcta. En caso de ser soluciones erróneas se facilitarán las indicaciones oportunas para su corrección.
- Registrar el número total de las soluciones enviadas por el alumno, además de otra información relevante para el curso, pudiendo esta ser consultada por los profesores de la asignatura.
- Generar reportes del desempeño académico de los estudiantes, tanto de manera individual como colectiva para obtener una mejor comprensión del avance y dificultades que tienen los estudiantes en los distintos temas.
- Desarrollar opciones de mantención al Sistema construido, gestión de ejercicios, información histórica y otros para obtener un ambiente amigable e intuitivo con el usuario y que sea de fácil mantención para los administradores.
- Probar el Software, siguiendo estándares establecidos en la ingeniería del software.

4.3 Descripción Global del Producto

4.3.1 Interfaz de usuario

La interfaz de usuario debe ser lo más amigable y familiar posible para reducir la resistencia al cambio y sea lo más adaptable posible, por lo que se contempla que tendrá lo siguiente:

- Título.
- Banner.
- Iconografía acorde a la acción a realizar (ingresar, realizar, editar, borrar, sugerencias).
- Iconografía representada por color a la acción a realizar (verde, rojo, amarillo, otros).
- Menú de navegación.

- Footer.
- La información se desplegará dentro de la misma página en los espacios disponibles.
- Menú desplegable.
- Iconos de información, desplegarán información cuando el usuario los pulse.

4.3.2 Interfaz De Hardware

Para la Universidad no es necesario comprar nuevos dispositivos para implementar el software ya que en la Facultad de Ciencias Empresariales ya trabaja y cuenta con laboratorios de computación equipados con equipos más que suficientes y modernos, el sistema se encontrará ubicado en un servidor X de la misma universidad, por esta razón no hay inconvenientes en cuanto a servidores o a la configuración de los equipos de los profesores.

Para los alumnos solo se debe contar con el hardware básico, mouse, teclado, monitor, ningún hardware que necesite ser configurado para utilizar el sistema, dado que se tratará de un sistema ubicado en servidores propios de la Universidad, por lo que ya se encuentra configurado de manera óptima para ser utilizados en los equipos disponibles para los alumnos, ya sean de la universidad cómo equipos propios.

4.3.3 Interfaz Software

El software se puede implementar sin la necesidad actual de agregar otro software específico, ya sea en el servidor disponible en la universidad, equipos del laboratorio o propios de los usuarios, siempre que tengan un software básico como Windows, Linux o alguna otra distribución de S.O. y un navegador para acceder a Internet.

4.3.4 Interfaces de comunicación

Las interfaces de comunicación para desarrollar el software y los procesos dentro el mismo son los siguientes protocolos:

HTTP: Es uno de los protocolos más utilizados de internet, ya que su principal función es la transferencia de archivos.

SOAP: (siglas de Simple Object Access Protocol) es un protocolo estándar que define como dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML.

HTML: Para la creación de páginas web. Utilizado para describir la estructura y el contenido en forma de texto de la página web o aplicación.

4.4 Requerimientos Específicos

4.4.1 Requerimientos Funcionales del sistema

Se presentan los requerimientos funcionales principales del sistema donde se destacan las principales características que poseerá, cada requerimiento representa a un módulo o opciones que el sistema presentará en su etapa final.

ID	Nombre	Descripción
1	Autenticar Usuario	EL sistema sólo puede ser accedido por usuarios registrados y con las credenciales correctas, en caso contrario mostrar mensaje de alerta o error
2	Crear Usuario	El sistema debe permitir la creación de nuevos usuarios sólo desde el módulo del administrador y profesor.
3	Modificar Usuario	El sistema debe permitir al administrador y profesor modificar los datos de los usuarios en caso de ingresar uno erróneo, luego de su creación
4	Eliminar Usuario	Quitar a los usuarios que ya no están activos tanto el sistema como de la base de datos.
5	Ingresar Problemas	Registrar los problemas en el sistema con sus datos, por ejemplo descripción, solución, enunciado.
6	Modificar Problemas	Editar algún dato del problema que requiera corrección.

7	Eliminar Problemas	Eliminar los problemas que se encuentran obsoletos o ingresaron por error.
8	Consultar Problemas	Consultar problemas en específico en la base de datos a través de un buscador en la página web.
9	Crear Cuaderno de ejercicios	Crear un cuaderno para cada alumno con problemas de diferentes tópicos y dificultades.
10	Modificar Cuaderno	Realizar cambios o editar un determinado cuaderno en el sistema.
11	Eliminar Cuaderno	Borrar un cuaderno determinado que ya no se use.
12	Consultar Cuaderno	Ejecutar consulta al sistema por un cuaderno específico mediante su id.
13	Resolver Problema	El alumno selecciona un problema de su cuaderno y propone una solución a dicho problema en el área disponible para escribir su código.
14	Evaluar Solución	El Sistema realiza una consulta a la base de datos con los parámetros ingresados por el alumno y devuelve el nivel de compatibilidad con los parámetros almacenados en la base de datos al momento de registrar la solución óptima al crear el problema.
15	Crear Curso	Crear un curso para los alumnos correspondientes.
16	Consultar curso	Ejecutar consulta al sistema por un curso específico mediante su id.
17	Eliminar Curso	Eliminar los cursos que se encuentran obsoletos o ingresaron por error.

18	Inscribe/Desinscribe Alumno	Inscribir o desinscribir un determinado alumno para un curso.
19	Enviar Problema Simple	Se envía un problema seleccionado por el profesor de manera global a los cuadernos de los alumnos.
20	Enviar Problema Aleatorio	Se envía un problema de forma aleatoria a los cuadernos de los alumnos.

Tabla 1: Requerimientos Funcionales

4.4.2 Interfaces externas de entrada

Cada interfaz de entrada indica todos los grupos de datos que serán ingresados al sistema independiente del medio de ingreso.

Identificador	Nombre del ítem.	Detalle de Datos contenidos en ítem
IDE_1	Login Usuario	Usuario, contraseña
IDE_2	Datos pregunta	pregunta, solución correcta, Tipo de ejercicio, Enunciado, Prototipo de función,
IDE_3	Feedback	retroalimentación de parte de los estudiantes a los desarrolladores
IDE_4	Envío de respuestas	respuestas del cuaderno de ejercicios

Tabla 2: interfaces externas de entrada.

4.4.3 Interfaces externas de Salida

Se especifica cada salida del sistema, indicando en cada caso el formato o medio de salida.

Identificado	Nombre del ítem.	Detalle de Datos contenidos en ítem	Medio Salida
IDS_1	Cuaderno de ejercicios	EJERCICIOS	Pantalla

Tabla 3: Interfaces externas de salida.

4.4.4 Atributos del producto

- **USABILIDAD- OPERABILIDAD.** Que los mensajes de advertencia sean claros y concisos para el usuario. Que en el 100% de los casos indique si se puede ingresar determinados caracteres o sólo los acepta de determinada manera, así mismo si olvida rellenar algún campo de texto .
- **FUNCIONALIDAD - SEGURIDAD:** El sistema deberá estar protegido a su totalidad, lo que se espera conseguir mediante una sesión online protegida por contraseña. La misma plataforma estará a cargo de asignar y administrar roles y permisos para delimitar los casos de uso dentro de la aplicación.
- **USABILIDAD - OPERATIVIDAD:** El módulo tendrá una interfaz fácil de utilizar y de entender, que ayude a los usuarios de forma que sea intuitiva en las funciones disponibles.
- **MANTENIMIENTO - FACILIDAD DE ANÁLISIS:** El sistema será fácil de entender, con códigos comentados para facilitar su comprensión y su posterior modificación.

5 FACTIBILIDAD

5.1 Factibilidad técnica.

Los instrumentos con los que cuenta el Departamento de Sistemas de Información permiten la correcta utilización del sistema de apoyo para la programación en C, sin requerir de licencias nuevas o software adicional.

El departamento cuenta con los siguientes recursos en su servidor:

- Sistema Operativo Linux (6)
- PHP (versión 5 o superior)
- MySQL

Los dispositivos que se encuentran en los laboratorios de la facultad responden de buena manera a las necesidades para el software de apoyo, sin la necesidad de adquirir software adicional.

Los laboratorios cuentan con los siguientes recursos:

- PROCESADOR (falta información)
- RAM (8 GB)
- DISCOS DUROS (falta información)
- Windows 10 (última versión)

Igualmente se posee el conocimiento, junto con las habilidades necesarias para la utilización correcta del software, lo que nos permite prescindir de cualquier tipo de capacitación.

5.2 Factibilidad operativa.

El principal impacto se encuentra en la facilitación del control en cómo los alumnos van avanzando a lo largo de la asignatura, permitiéndole al docente poder reforzar los contenidos donde más errores hayan habido. Junto con esto también se encuentra la información de tiempo que cada estudiante le dedica a la asignatura en función de ejercicios que haya realizado de los cuadernos de ejercicios generados y una disminución de tiempo que el docente tendrá que dedicarle a la asignatura ahorrando tiempos en revisión de actividades. Se espera que tenga un efecto no menor en el impacto que tienen los estudiantes en su primer acercamiento a la programación que muchas veces puede terminar en un abandono prematuro de la carrera.

El alumno tendrá la posibilidad de realizar ejercicios de los contenidos vistos en clases y obtener un feedback inmediato para conocer su desempeño, con acceso a la respuesta propuesta y a realizar de nuevo la actividad si no logró completarla exitosamente al primer intento.

Cada cuaderno de ejercicios será único por alumno, en busca de ayudar al docente en la disminución de copias entre los mismo estudiantes.

Al mismo tiempo cabe recalcar que no será necesario que los alumnos o encargados de laboratorio tengan que realizar una instalación del software o algo similar, dado que al tratarse de un software pensado para la web sólo necesitarán conocer el link de acceso a la plataforma. De la misma manera se establece que los mismos profesores tengan acceso a privilegios de administrador para sus cursos y puedan realizar los cambios necesarios en su asignatura y los datos de sus estudiantes de manera más rápida, también se tiene estimado que exista un usuario registrado con derechos de superadministrador el cual puede ser un profesor o alguien designado por el departamento de SI, para que cumpla este rol y pueda actualizar y modificar la información disponible de alumnos, profesores, problemas, cursos y otros datos de la plataforma.

5.3 Factibilidad económica.

La automatización del proceso de ejercitación posee ventajas evidentes para ambas partes, tanto para los alumnos como para los docentes. Para la implementación ya se cuenta con todas las herramientas necesarias, prescindiendo de recursos económicos para adquirir algún tipo de hardware o software ya que no se hará uno tampoco de ninguna licencia, limitándose a usar software libres.

Para una correcta cuantificación de gastos se debe considerar el desarrollo del software mismo, y los gastos asociados, que se detallarán a continuación.

- energía, luz(Se estima un gasto de 10.000 mensual en electricidad)
- internet(Se estima un plan de internet de 40.000 mensual)
- Horas de trabajo(Se considera un total de 25 a 30 horas hombre semanales)

En el caso de las horas hombres se establece un precio estimado de \$4000 por hora trabajada, trabajando entre 25 a 30 horas semanales, obteniendo un promedio de 27 horas HH a la semana durante 4 meses(estimando un total de 17 semanas) en el desarrollo de la aplicación.

Tarea	Horas hombre	Duración en meses	Precio hora hombre	Participantes	Total
Desarrollo de software	27 hrs.	4	\$4000	2	\$3672000

Tabla 4: Gastos estimados proyecto.

Por lo tanto el costo asociado al desarrollo de la aplicación se estiman en unos \$3672000 durante la producción del proyecto en los primeros meses, se espera que finalizada la etapa de producción se reduzcan las horas de desarrollo y mantenimiento, dado que se enfocara principalmente en sacar pequeñas actualizaciones y corregir errores o problemas pequeños que se puedan ir presentando, de esta forma se estima que las HH bajen a un promedio de 10 horas a la semana a partir del quinto mes (manteniéndose la misma cantidad de meses y precio hora hombre).

Los gastos fijos considerados durante el desarrollo de este proyecto fueron los siguientes:

Item	Costo	Tiempo en meses	Costo total
Luz	\$10000	4	\$40000
Internet	\$40000	4	\$160000
Total costos fijos			\$200000

Tabla 5: Gastos de electricidad y conectividad.

Por consiguiente el costo asociado a los gastos fijos se estima en un total de \$200000 durante la producción del software. El costo del arriendo o compra de un servidor no se consideró, dado que la universidad ya consta con servidores disponibles para albergar la aplicación, dejando el total de gastos en \$3872000.

De acuerdo a los beneficios otorgados a la facultad y los dispositivos que esta cuenta, se estima que existe un ahorro de tiempo al implementar el software y así reducir gastos de tiempo en los docentes y alumnos.

En conclusión de acuerdo a los beneficios que otorga el desarrollo del proyecto y considerando que este proyecto ayuda a los alumnos a disminuir la tasa de reprobación de las asignaturas de programación en lenguaje C y que usan otros lenguajes basados en su lógica, lo que conlleva a que no se atrasen en la carrera, lo cual significa un ahorro para ellos, lo que produce un beneficio considerable ayudando a tener una mayor tasa de retención de estudiantes en los 1eros años, es que se dice que la factibilidad económica de la Facultad es viable.

5.4 Conclusión de la factibilidad

Mediante el análisis realizado se demuestra las necesidades que existen en las diferentes asignaturas relacionadas con el lenguaje de desarrollo en C, de crear un sistema que les facilite la comunicación con los alumnos para poder realizar la asignación de ejercicios y a la vez poder llevar un monitoreo del avance de estos mismo en las asignaturas correspondientes.

En resumen a todos los análisis ya mencionados en este capítulo se concluye que el proyecto es factible de desarrollar , ya que no tiene costos de implementación y desarrollo, a su vez entrega beneficios importantes a todos los usuarios que utilizan el sistema, colaborando de esta manera en el desarrollo de las carreras que imparten asignaturas en las cuales los alumnos deben trabajar sus códigos y clases prácticas con lenguaje C.

6 ANÁLISIS

6.1 Procesos de Negocios futuros

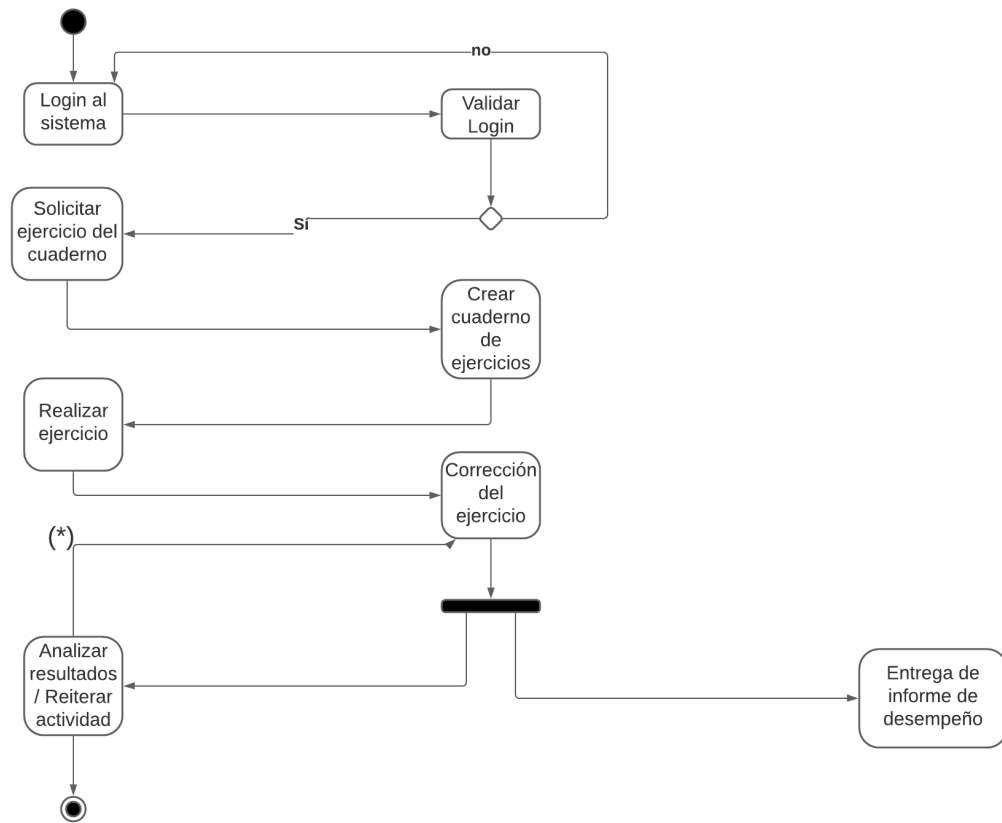


Figura 6.1.1: Diagrama de BPMN

Mediante el diagrama bpmn se puede observar el flujo de trabajo que los Estudiantes emplearían para la utilización del software de ejercitación en línea. En el proceso de “Analizar resultados / Reiterar Actividad” se incluye el caso de que el estudiante no haya completado el cuaderno de manera satisfactoria y quiera reiterar la actividad, lo anteriormente mencionado hace alusión al flujo marcado con un (*).

6.2 Diagrama de Flujo de Datos

A continuación se mostrará mediante imágenes los diagramas de flujo de datos en sus 2 niveles contexto y superior respectivamente, junto con una breve descripción sobre cada flujo de relación, con él fin de dar apoyo a los casos de uso, y un mejor entendimiento de cómo el trasfondo del sistema ejecuta las distintas actividades y peticiones realizadas por los usuarios durante su uso, junto con los datos que son enviados y recibidos para cada petición realizada tanto al backend como a la base de datos, de forma de que sea más claro el funcionamiento interno de la plataforma.

Diagrama de flujo de datos: Nivel de contexto

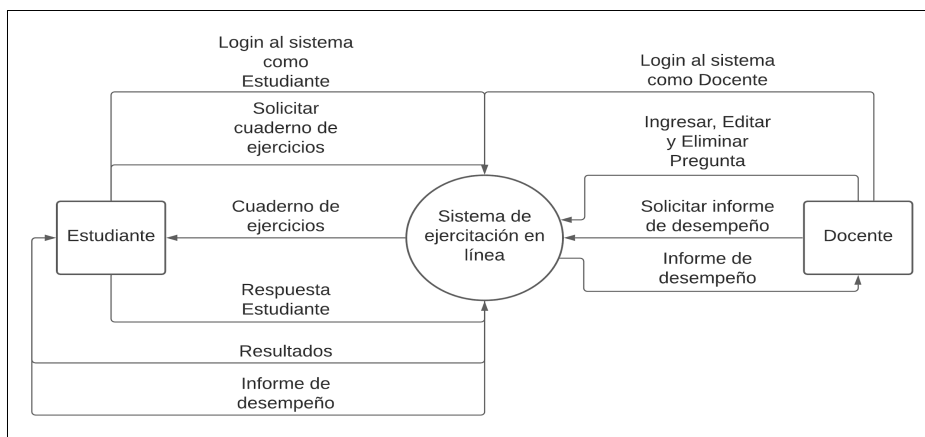


Figura 6.2.1: Diagrama de flujo de datos nivel de contexto

Diagrama de flujo de datos: Nivel superior

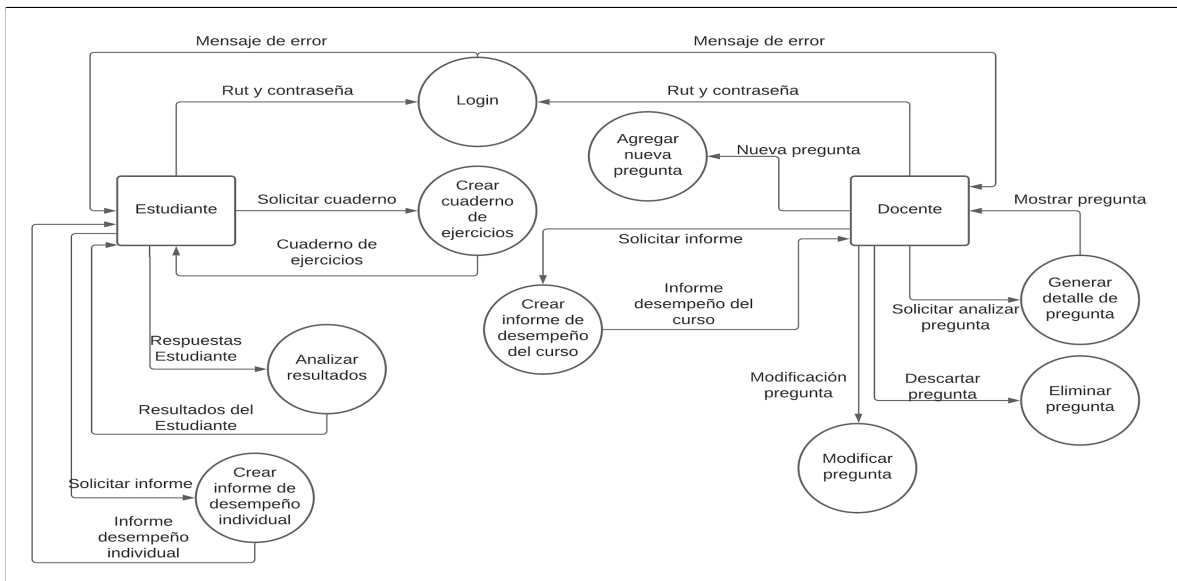


Figura 6.2.2: Diagrama de flujo de datos de nivel superior

6.3 Diagrama de casos de uso

En este apartado se mostrará cada uno de los requerimientos funcionales presentados en los capítulos anteriores.

6.3.1 Actores

Estudiante: su único requisito es contar con una cuenta vigente en la red Intranet UBB. Usuario corresponde al nivel de privilegio más bajo en el software y solo cuenta con acceso a las funciones de ingreso de sesión, solicitar cuaderno de ejercicios y envío de las respuestas.

Profesor: Profesor tiene un privilegio mayor a los estudiantes, ya que encima de lo anteriormente mencionado puede también agregar preguntas al repositorio de preguntas si así lo encuentra necesario.

Administrador: El Administrador cumple con el rol de mantención y limpieza del software, tiene acceso a la totalidad del sistema y a todas sus opciones puede agregar, eliminar, modificar tanto a usuarios como profesores de los distintos ramos, así también como los distintos cuadernos de ejercicios para cada uno.

Sistema Web: encargado de compilar y ejecutar el código proporcionado por los alumnos y entregar una respuesta, utilizando API proporcionada por la web.

6.3.2 Casos de Uso y descripción

Como puede verse en el diagrama de casos de uso, se presentan las interacciones de diferentes usuarios que tiene el sistema.

Los actores que son el administrador, profesores y alumnos interactúan con el sistema online por medio de un computador con acceso a internet y la plataforma, en donde deben ingresar con sus credenciales de inicio de sesión para tener acceso al sistema, luego de que sean autenticados por el mismo, luego de esto pueden acceder a las diferentes tareas y módulos disponibles según su nivel de usuario.

El actor llamado sistema web(en este caso el sistema), es el que se encarga de compilar y ejecutar códigos fuente en el lenguaje C ingresados por los estudiantes como solución a problemas sugeridos por los profesores.

Diagramas de casos de uso:

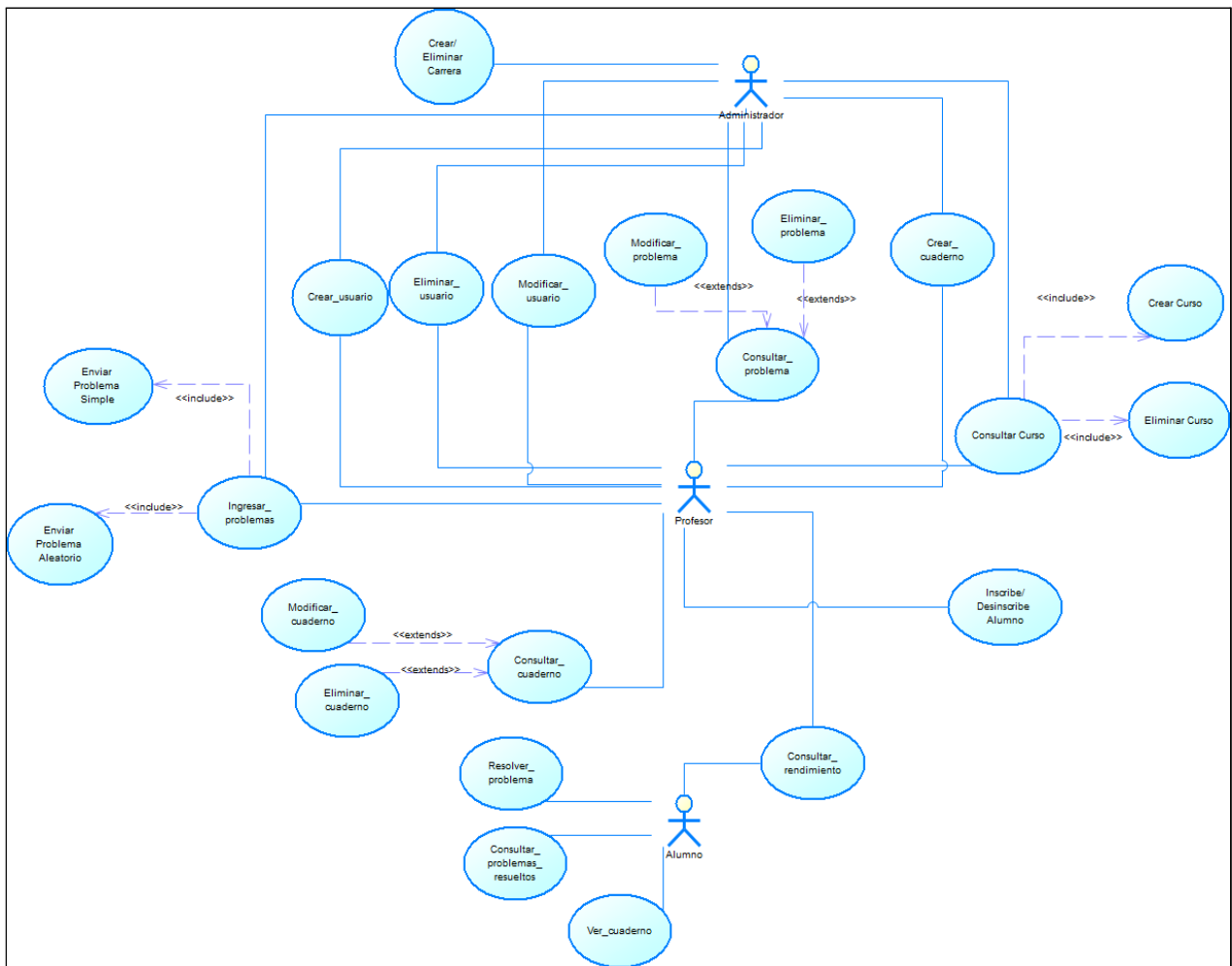


Figura 6.3.2.1: Casos de uso general, actores principales.

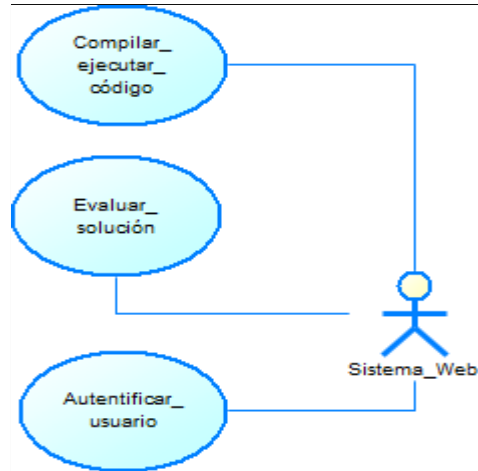


Figura 6.3.2.2: Casos de uso del sistema web

6.3.3 Especificación de los Casos de Uso

6.3.3.1 Caso de Uso: <Autenticar Usuario>

- Descripción: La plataforma devuelve al usuario autenticado
- Pre-Condiciones:
 - El usuario debe iniciar sesión
- Flujo de Eventos Básicos:

Al actor	El sistema
	1.- Se conecta a la base de datos para obtener datos de usuario
	2.- Muestra perfil según autenticación de usuario.

- Flujo de Eventos Alternativo:

Al actor	El sistema
	2.1.- Se niega el acceso al sistema con un mensaje de error.

- Post-Condiciones:

- Usuario autenticado en el sistema.

6.3.3.2 Caso de Uso: <Ingresar Problemas>

- Descripción: Se ingresan los problemas para que los alumnos puedan visualizarlos desde sus cuadernos de ejercicios, estos son almacenados en la base de datos y sólo los usuarios registrados como profesores o administradores tienen acceso a esta opción.
- Pre-Condiciones:
 - El usuario debe iniciar sesión.
 - Estar logueado como profesor o administrador.
 - Que exista al menos un usuario profesor o administrador en el sistema.
 - Flujo de Eventos Básicos:

Al actor	El sistema
1.- Ingresar el problema y los atributos correspondientes a este.	2.- Validar los datos del problema ingresado
	3.- Verificar los datos del problema ingresado.
	4.- Guardar en la base de datos el problema ingresado

- Flujo de Eventos Alternativo:

Al actor	El sistema
3(a)	3.1.- Los datos ingresados no son admitidos por el sistema.
4(a)	4.1.- Los datos y el problema ingresado no son guardados, por no ser admitidos o un fallo de conexión.

- Post-Condiciones:
 - Se muestra un mensaje de éxito en el ingreso del problema.

6.3.3.3 Caso de Uso: <Consultar Problema>

- Descripción: Los usuarios que ingresan los problemas también pueden visualizarlos para evitar problemas duplicados en la base de datos y saber cuales ya están ingresados
- Pre-Condiciones:
 - Estar logueado como profesor o administrador.
 - Existencia de al menos un problema en la base de datos.
- Flujo de Eventos Básicos:

Al actor	El sistema
1.- Selecciona el problema que quiere visualizar.	2.- Conecta con la base de datos y se muestran los problemas de acuerdo al seleccionado

- Flujo de Eventos Alternativo:

Al actor	El sistema
	2.1.- Falla en la conexión y error en la consulta

- Post-Condiciones:
 - Se muestra la respuesta a la consulta realizada a la base de datos.

6.3.3.4 Caso de Uso: <Modificar Problema>

- Descripción: Al realizar una modificación de un problema se consulta por la existencia de este, luego de esto se pueden actualizar los datos (es un extends del caso de uso consultar problema, lo utiliza para comprobar la existencia del problema a actualizar).
- Pre-Condiciones:
 - Deben existir problemas en la base de datos.
 - Estar logeado como administrador o profesor.
- Flujo de Eventos Básicos:

Al actor	El sistema
1.- Selecciona la opción consultar problema	2.- Invoca al caso de uso consultar problemas.
3.- Selecciona el campo a modificar	4.- Muestra el campo a modificar .
	5.- valida los datos nuevos.
	6.- Verifica los datos nuevos del problema
	7.- Guarda las modificaciones del problema en la base de datos.

- Flujo de Eventos Alternativo:

Al actor	El sistema
4(a)	4.1.- El campo seleccionado no se puede modificar, muestra mensaje de alerta
5(a)	5.1.- Los datos a ingresar no son válidos, muestra mensaje error

- Post-Condiciones:
 - El problema se actualiza en la base de datos.

6.3.3.5 Caso de Uso: <Eliminar Problema>

- Descripción: Se elimina un problema de la base de datos, pero primero se consulta por la existencia de este, luego de esto se puede eliminar (es un extends del caso de uso consultar problema, lo utiliza para comprobar la existencia del problema a eliminar).
- Pre-Condiciones:
 - Deben existir problemas en la base de datos.
 - Estar logeado como administrador o profesor.
- Flujo de Eventos Básicos:

Al actor	El sistema
1.- Selecciona la opción consultar problema.	2.- Invoca el caso de uso consultar problema.
3.- Selecciona el problema a eliminar.	4.- Elimina el problema de la base de datos, muestra un mensaje de éxito.

- Flujo de Eventos Alternativo:

Al actor	El sistema
4(a)	4.1.- Fallo en eliminar problema de la base de datos, muestra mensaje de error.

- Post-Condiciones:
 - El problema se elimina de la base de datos.

6.3.3.6 Caso de Uso: <Crear Cuaderno>

- Descripción: Se ingresan los datos para crear un cuaderno, estos son almacenados en la base de datos y sólo los usuarios registrados como administradores o profesores tienen acceso a esta opción.
- Pre-Condiciones:
 - El usuario debe iniciar sesión.
 - Estar logueado como administrador o profesor.
 - Que exista al menos un administrador o profesor en el sistema.
- Flujo de Eventos Básicos:

Al actor	El sistema
1.- Ingresar el cuaderno y los atributos correspondientes a este.	2.- Validar los datos del cuaderno ingresado y crearlo automáticamente al realizar una inscripción.
	3.- Verificar los datos del cuaderno ingresado.
	4.- Guardar en la base de datos el cuaderno ingresado.

- Flujo de Eventos Alternativo:

Al actor	El sistema
3(a)	3.1.- Los datos ingresados no son admitidos por el sistema.
4(a)	4.1.- Los datos y el cuaderno ingresado no son guardados, por no ser admitidos o un fallo de conexión.

- Post-Condiciones:
 - Se muestra un mensaje de éxito en el ingreso del cuaderno.

6.3.3.7 Caso de Uso: <Consultar Cuaderno>

- Descripción: Los usuarios que ingresan los cuadernos también pueden visualizarlos para evitar problemas de cuadernos duplicados en la base de datos y saber cuales ya están ingresados
- Pre-Condiciones:
 - Estar logeado como profesor o administrador.
 - Existencia de al menos un cuaderno.
- Flujo de Eventos Básicos:

Al actor	El sistema
1.- Selecciona el cuaderno que quiere visualizar.	2.- Conecta con la base de datos y se muestran los datos de acuerdo al cuaderno seleccionado y el rut del usuario.

- Flujo de Eventos Alternativo:

Al actor	El sistema
2(a)	2.1.- Falla en la conexión y error en la consulta

- Post-Condiciones:
 - Se muestra la respuesta a la consulta realizada a la base de datos.

6.3.3.8 Caso de Uso: <Modificar Cuaderno>

- Descripción: Al realizar una modificación de un cuaderno se consulta por la existencia de este, luego de esto se pueden actualizar los datos (modificar cuaderno es un extends del caso de uso consultar cuaderno, lo utiliza para comprobar la existencia del cuaderno a actualizar).
- Pre-Condiciones:
 - Deben existir cuadernos en la base de datos.
 - Estar logeado como administrador o profesor.
- Flujo de Eventos Básicos:

Al actor	El sistema
1.- Selecciona la opción consultar cuaderno	2.- Invoca al caso de uso consultar cuaderno.
3.- Selecciona el campo a modificar	4.- Muestra el campo a modificar .
	5.- valida los datos nuevos.
	6.- Verifica los datos nuevos del cuaderno
	7.- Guarda las modificaciones del cuaderno en la base de datos.

- Flujo de Eventos Alternativo:

Al actor	El sistema
4(a)	4.1.- El campo seleccionado no se puede modificar, muestra mensaje de alerta
5(a)	5.1.- Los datos a ingresar no son válidos, muestra mensaje error

- Post-Condiciones:
 - El cuaderno se actualiza en la base de datos.

6.3.3.9 Caso de Uso: <Eliminar Cuaderno>

- Descripción: Se elimina un cuaderno de la base de datos, pero primero se consulta por la existencia de este, luego de esto se puede eliminar (eliminar cuaderno es un extends del caso de uso consultar cuaderno, lo utiliza para comprobar la existencia del cuaderno a eliminar).
- Pre-Condiciones:
 - Deben existir cuadernos en la base de datos.
 - Estar logueado como administrador o profesor.
- Flujo de Eventos Básicos:

Al actor	El sistema
1.- Selecciona la opción consultar cuaderno, alumno o curso.	2.- Invoca el caso de uso consultar consultar correspondiente a la consulta pertinente(cuaderno, alumno, curso).
3.- Selecciona el dato a eliminar.	4.- Elimina el cuaderno de la base de datos de forma automática, muestra un mensaje de éxito.

- Flujo de Eventos Alternativo:

Al actor	El sistema
4(a)	4.1.- Fallo en eliminar cuaderno de la base de datos, muestra mensaje de error.

- Post-Condiciones:
 - El cuaderno se elimina de la base de datos.

6.3.3.10 Caso de Uso: <Crear Usuario>

- Descripción: Se ingresan los datos para crear un usuario, estos son almacenados en la base de datos y sólo los usuarios registrados como profesores o administrador tienen acceso a esta opción.
- Pre-Condiciones:
 - El usuario debe iniciar sesión.
 - Estar logueado como profesor o administrador.
 - Que exista al menos un profesor en el sistema.
- Flujo de Eventos Básicos:

Al actor	El sistema
1.- Ingresar el usuario y los atributos correspondientes a este.	2.- Validar los datos del usuario ingresado
	3.- Verificar los datos del usuario ingresado.
	4.- Guardar en la base de datos al usuario ingresado

- Flujo de Eventos Alternativo:

Al actor	El sistema
3(a)	3.1.- Los datos ingresados no son admitidos por el sistema.
4(a)	4.1.- Los datos y el usuario ingresado no son guardados, por no ser admitidos o un fallo de conexión.

- Post-Condiciones:
 - Se muestra un mensaje de éxito en el ingreso del usuario.

6.3.3.11 Caso de Uso: <Modificar Usuario>

- Descripción: Al realizar una modificación de un usuario se consulta por la existencia de este, luego de esto se pueden actualizar los datos.
- Pre-Condiciones:
 - Deben existir usuarios en la base de datos.
 - Estar logueado como profesor o administrador.
- Flujo de Eventos Básicos:

Al actor	El sistema
1.- Selecciona la opción modificar usuario	2.- Invoca al caso de uso modificar usuario.
3.- Selecciona el campo a modificar	4.- Muestra el campo a modificar .
	5.- valida los datos nuevos.
	6.- Verifica los datos nuevos del usuario
	7.- Guarda las modificaciones del usuario en la base de datos.

- Flujo de Eventos Alternativo:

Al actor	El sistema
4(a)	4.1.- El campo seleccionado no se puede modificar, muestra mensaje de alerta
5(a)	5.1.- Los datos a ingresar no son válidos, muestra mensaje error

- Post-Condiciones:
 - El usuario se actualiza en la base de datos.

6.3.3.12 Caso de Uso: <Eliminar Usuario>

- Descripción: Se elimina un usuario de la base de datos, pero primero se consulta por la existencia de este, luego de esto se puede eliminar.
- Pre-Condiciones:
 - Deben existir usuarios en la base de datos.
 - Estar logueado como administrador o profesor.
- Flujo de Eventos Básicos:

Al actor	El sistema
1.- Selecciona la opción eliminar usuario.	2.- Invoca el caso de uso eliminar usuario
3.- Selecciona el usuario a eliminar.	4.- Elimina el usuario de la base de datos, muestra un mensaje de éxito.

- Flujo de Eventos Alternativo:

Al actor	El sistema
4(a)	4.1.- Fallo en eliminar al usuario de la base de datos, muestra mensaje de error.

- Post-Condiciones:
 - El usuario se elimina de la base de datos.

6.3.3.13 Caso de Uso: <Consultar Rendimiento>

- Descripción: Sección para informar el rendimiento de los usuarios del curso o por cada alumno, dependiendo el usuario que realice la consulta.
- Pre-Condiciones:
 - El usuario debe encontrarse autenticado.
 - Deben existir cuadernos ingresados en el sistema para el alumno o alumnos a consultar.
- Flujo de Eventos Básicos:

Al actor	El sistema
1.- Selecciona la opción consultar rendimiento en el menú.	2.- Se conecta a la base de datos regresando un listado con los alumnos y el porcentaje de resolución por tipo de problemas.
	3.- Emite el informe de rendimiento grupal en formato PDF.

- Flujo de Eventos Alternativo:

Al actor	El sistema
1(a).- Selecciona a un alumno en específico	4.- Se conecta a la base de datos y regresa el progreso del alumno y la calidad de sus soluciones ingresadas.

- Post-Condiciones:
 - El usuario puede ver su informe de rendimiento por pantalla.

6.3.3.14 Caso de Uso: <Ver Cuaderno>

- Descripción: El sistema realiza una consulta y obtiene el cuaderno de ejercitación personal del alumno.
- Pre-Condiciones:
 - Estar logueado como alumno.
- Flujo de Eventos Básicos:

Al actor	El sistema
1.- El alumno selecciona la opción ver cuaderno en su menú.	2.- Se conecta a la base de datos y regresa una lista de los problemas asociados al rut e id del cuaderno del alumno.

- Flujo de Eventos Alternativo:

Al actor	El sistema
	2.1.- Fallo en la conexión o consulta, no se visualizarán los problemas

- Post-Condiciones:
 - El alumno puede visualizar su cuaderno de ejercitación.

6.3.3.15 Caso de Uso: <Resolver problema>

- Descripción: Muestra al usuario alumno el enunciado y los parámetros de un problema y el espacio para poder desarrollarlo. .
- Pre-Condiciones:
 - Estar logueado como alumno.
 - Debe existir al menos un problema en el sistema.
 - Debe contar con su cuaderno de ejercicios en el sistema.
 - Haber seleccionado la opción consultar cuaderno.
- Flujo de Eventos Básicos:

Al actor	El sistema
1.- El usuario selecciona el problema en su cuaderno.	2.- Se conecta a la base de datos y regresa el enunciado y parámetros del problema.
	3.- Actualiza la pantalla con el ambiente para resolver el problema.
4.- El usuario ingresa su código fuente en lenguaje C.	
5.- Presiona el botón “enviar”	6.- Muestra mensaje de éxito “solución registrada”.

- Flujo de Eventos Alternativo:

Al actor	El sistema
4(a).- El usuario no ingresa su código fuente	7.- No realiza ninguna operación, ni considera el intento cómo malo.

- Post-Condiciones:
 - La solución se ingresa al sistema.
 - El alumno recibe la calidad de la solución que envió.

6.3.3.16 Caso de Uso: <Procesar Respuesta del Alumno>

- Descripción: Compila y ejecuta el código fuente en lenguaje C enviado por algún alumno.
- Pre-Condiciones:
 - Estar logueado como alumno.
 - Debe existir un cuaderno de ejercicios disponible en el sistema.
 - El usuario alumno debe haber ingresado a la opción resolver problema y haber enviado su código fuente.
- Flujo de Eventos Básicos:

Al actor	El sistema
	1.- Se conecta con una API a un sistema web
	2.- Compila y ejecuta el código fuente enviado por el sistema con los casos de pruebas enviados.

- Flujo de Eventos Alternativo:

Al actor	El sistema
	1.1.- Se le informa al usuario problemas de conexión o ejecución mediante un mensaje de error.

- Post-Condiciones:
 - El sistema queda listo para emitir la calidad de la solución ingresada.

6.3.3.17 Caso de Uso: <Evaluar Solución>

- Descripción: Compila y ejecuta el código fuente proporcionado por el alumno.
- Pre-Condiciones:
 - Estar logueado como alumno.
 - Debe existir un cuaderno disponible en el sistema.
 - El usuario alumno debe haber ingresado alguna solución.
- Flujo de Eventos Básicos:

Al actor	El sistema
	1.- Solicita a un web service ejecutar el problema con los distintos casos de pruebas ingresados.
2.- Ejecuta código enviado por el sistema con sus respectivos casos de pruebas y entrega las salidas.	3.- Compara las salidas entregadas por el servicio web con las salidas esperadas disponibles en la base de datos.
	4.- Registra la compatibilidad entre las salidas esperadas idóneas y las salidas entregadas por el servicio web.

- Flujo de Eventos Alternativo:

Al actor	El sistema
2(a).- Informa de algún error al sistema	3.1.- Informa al usuario el error ocurrido mediante un mensaje de alerta.

- Post-Condiciones:
 - Se registra en el sistema la calidad de solución del problema .

6.3.3.18 Caso de Uso: <Consultar Problemas Resueltos>

- Descripción: Permite a los usuarios(alumnos) visualizar los problemas que ya fueron resueltos por ellos.
- Pre-Condiciones:
 - Estar logueado en el sistema como alumno.
 - Debe existir problemas asignado a un alumno.
 - El alumno debe haber ingresado una solución como mínimo al problema.
- Flujo de Eventos Básicos:

Al actor	El sistema
1.- El usuario selecciona la opción el menú consultar problemas resueltos.	2.- Se conecta a la base de datos regresando un listado con los problemas resueltos por el alumno que hizo la consulta.

- Flujo de Eventos Alternativo:

Al actor	El sistema
	2.1.- Falla en la conexión con la base de datos, se muestra un mensaje de error “imposible conectar con BDD en este momento”.

- Post-Condiciones:
 - El usuario obtiene el listado de problemas resueltos.

6.3.3.19 Caso de Uso: <Consultar Curso>

- Descripción: El sistema realiza una consulta y obtiene los cursos inscritos por el usuario.
- Pre-Condiciones:
 - Estar logueado.
- Flujo de Eventos Básicos:

Al actor	El sistema
1.- El usuario selecciona la opción ver cuaderno en su menú.	2.- Se conecta a la base de datos y regresa una lista de los cursos asociados al rut del usuario.

- Flujo de Eventos Alternativo:

Al actor	El sistema
	2.1.- Fallo en la conexión o consulta, no se visualizarán los cursos

- Post-Condiciones:
 - El usuario puede visualizar sus cursos.

6.3.3.20 Caso de Uso: <Crear Curso>

- Descripción: Se ingresan los datos para crear un curso, estos son almacenados en la base de datos y sólo los usuarios registrados como profesores o administradores tienen acceso a esta opción.
- Pre-Condiciones:
 - El usuario debe iniciar sesión.
 - Estar logueado como profesor o administrador.
 - Que exista al menos un profesor en el sistema.
- Flujo de Eventos Básicos:

Al actor	El sistema
1.- Ingresar el curso y los atributos correspondientes a este.	2.- Validar los datos del curso ingresado.
	3.- Verificar los datos del curso ingresado.
	4.- Guardar en la base de datos el curso ingresado

- Flujo de Eventos Alternativo:

Al actor	El sistema
3(a)	3.1.- Los datos ingresados no son admitidos por el sistema.
4(a)	4.1.- Los datos y el curso ingresado no son guardados, por no ser admitidos, un fallo de conexión o un duplicado de datos.

- Post-Condiciones:
 - Se puede visualizar el curso correctamente.

6.3.3.21 Caso de Uso: <Eliminar Curso>

- Descripción: Se elimina un curso de la base de datos, pero primero se consulta por la existencia de este, luego de esto se puede eliminar (eliminar curso es un extends del caso de uso consultar cuaderno, lo utiliza para comprobar la existencia del curso a eliminar de la base de datos).
- Pre-Condiciones:
 - Deben existir cursos en la base de datos.
 - Estar logueado como administrador o profesor.
- Flujo de Eventos Básicos:

Al actor	El sistema
1.- Selecciona la opción consultar o ver cursos.	2.- Invoca el caso de uso consultar curso correspondiente a la consulta pertinente.
3.- Selecciona el curso a eliminar.	4.- Elimina el curso de la base de datos de forma automática, muestra un mensaje de éxito.
	5.- Elimina los cuadernos ligados al curso de manera automática.

- Flujo de Eventos Alternativo:

Al actor	El sistema
4(a)	4.1.- Fallo en eliminar curso de la base de datos, muestra mensaje de error.

- Post-Condiciones:
 - El curso se elimina de la base de datos y ya no puede ser visualizado.

6.3.3.22 Caso de Uso: <Inscribir Alumno>

- Descripción: Se selecciona el alumno a inscribir en el curso, estos son almacenados en la base de datos y son ligados a un curso. Solo los usuarios registrados cómo profesores tienen acceso a esta opción.
- Pre-Condiciones:
 - El usuario debe iniciar sesión.
 - Estar logueado como profesor o administrador en caso de ser importación.
 - Que exista al menos un profesor en el sistema.
- Flujo de Eventos Básicos:

Al actor	El sistema
1.- Selecciona al alumno o alumnos a inscribir en él curso.	2.- Valida los datos de los alumnos seleccionados.
	3.- Verifica los datos del alumno seleccionado.
	4.- Vincula el alumno al curso a través de una id

- Flujo de Eventos Alternativo:

Al actor	El sistema
3(a)	3.1.- El alumno seleccionado no es admitido por el sistema para ese curso.
4(a)	4.1.- Los datos del alumno y el curso ingresado no son guardados, por no ser admitidos, un fallo de conexión o un duplicado de datos.

- Post-Condiciones:
 - Se puede visualizar correctamente al alumno en la lista del curso.
 - El alumno tiene acceso a los cuadernos de ejercicios del curso.

6.3.3.23 Caso de Uso: <Desinscribir Alumno>

- Descripción: Se selecciona el alumno a desinscribir del curso, estos son eliminados de la conexión en la base de datos a dicho curso. Solo los usuarios registrados como profesores o administradores tienen acceso a esta opción.
- Pre-Condiciones:
 - El usuario debe iniciar sesión.
 - Estar logueado como profesor o administrador.
 - Que exista al menos un profesor en el sistema.
- Flujo de Eventos Básicos:

Al actor	El sistema
1.- Selecciona al alumno o alumnos a desinscribir en el curso.	2.- Valida los datos de los alumnos seleccionados.
	3.- Verifica los datos del alumno seleccionado.
	4.- Desvincula el alumno al curso a través de una consulta con su id

- Flujo de Eventos Alternativo:

Al actor	El sistema
3(a)	3.1.- El alumno seleccionado no puede ser desinscrito, muestra un mensaje de error.

- Post-Condiciones:
 - Se puede visualizar correctamente que el alumno no está en la lista del curso.
 - El alumno no tiene puede visualizar el curso en su pestaña de cursos.

6.3.3.24 Caso de Uso: <Enviar Problema Simple>

- Descripción: Se selecciona un problema de una lista para enviar a los cuadernos de los alumnos de forma manual por el profesor o administrador en su defecto.
- Pre-Condiciones:
 - El usuario debe iniciar sesión.
 - Estar logueado como profesor o administrador.
 - Que exista al menos un profesor en el sistema.
- Flujo de Eventos Básicos:

Al actor	El sistema
1.- Selecciona el problema o problemas a enviar.	2.- Valida los datos de los problemas seleccionados.
	3.- Verifica los datos de los problemas seleccionados.
	4.- Envía los problemas a los cuadernos correspondientes a la id del curso y cuaderno

- Flujo de Eventos Alternativo:

Al actor	El sistema
3(a)	3.1.- El problema seleccionado no se pudo enviar, muestra un mensaje de error.

- Post-Condiciones:
 - Se puede visualizar correctamente el problema en los cuadernos de los alumnos.

6.3.3.25 Caso de Uso: <Enviar Problema Aleatorio>

- Descripción: Se selecciona un problema de forma aleatoria de una lista para enviar, para posteriormente enviar a los cuadernos de los alumnos por el profesor o administrador si se le solicita. Solo los usuarios registrados como profesores o administradores tienen acceso a esta opción.
- Pre-Condiciones:
 - El usuario debe iniciar sesión.
 - Estar logueado como profesor o administrador.
 - Que exista al menos un profesor en el sistema.
- Flujo de Eventos Básicos:

Al actor	El sistema
1.-Selecciona enviar problema de forma aleatoria.	2.- Valida los datos de los problemas escogidos aleatoriamente.
	3.- Verifica los datos de los problemas.
	4.- Envía los problemas a los cuadernos correspondientes a la id del curso y cuaderno de los alumnos.

- Flujo de Eventos Alternativo:

Al actor	El sistema
3(a)	3.1.- El problema aleatorio no se pudo enviar, muestra un mensaje de error.

- Post-Condiciones:
 - Se puede visualizar correctamente el problema en los cuadernos de los alumnos.

6.3.3.26 Caso de Uso: <Crear Carrera>

- Descripción: Se ingresan los datos para crear una carrera, estos son almacenados en la base de datos y sólo los usuarios registrados como administrador tienen acceso a esta opción.
- Pre-Condiciones:
 - El usuario debe iniciar sesión.
 - Estar logueado como administrador.
 - Que exista al menos un administrador en el sistema.
- Flujo de Eventos Básicos:

Al actor	El sistema
1.- Ingresar la carrera y los atributos correspondientes a esta.	2.- Validar los datos de la carrera ingresada.
	3.- Verificar los datos de la carrera ingresada.
	4.- Guardar en la base de datos la carrera ingresada.

- Flujo de Eventos Alternativo:

Al actor	El sistema
3(a)	3.1.- Los datos ingresados no son admitidos por el sistema.
4(a)	4.1.- Los datos y la carrera ingresada no son guardados, por no ser admitidos, un fallo de conexión o un duplicado de datos.

- Post-Condiciones:
 - Se puede visualizar la carrera correctamente.

6.3.3.27 Caso de Uso: <Eliminar Carrera>

- Descripción: Se elimina una carrera de la base de datos, pero primero se consulta por la existencia de esta, luego de esto se puede eliminar.
- Pre-Condiciones:
 - Deben existir cursos en la base de datos.
 - Estar logueado cómo administrador.
- Flujo de Eventos Básicos:

Al actor	El sistema
1.- Selecciona la opción consultar o ver datos de la carrera.	2.- Invoca la consulta pertinente desde el back-end.
3.- Selecciona la carrera a eliminar.	4.- Elimina la carrera de la base de datos de forma automática, muestra un mensaje de éxito.
	5.- Elimina los datos ligados a la carrera de manera automática.

- Flujo de Eventos Alternativo:

Al actor	El sistema
4(a)	4.1.- Fallo en eliminar la carrera de la base de datos, muestra un mensaje de error.

- Post-Condiciones:
 - La carrera se elimina de la base de datos y ya no puede ser visualizada.

6.4 Modelamiento de datos

Los problemas son registrados en el sistema por los profesores o administradores que pueden ser los mismos, para que sean los alumnos los que deban resolver un conjunto de problemas, que se van registrando en sus cuadernos.

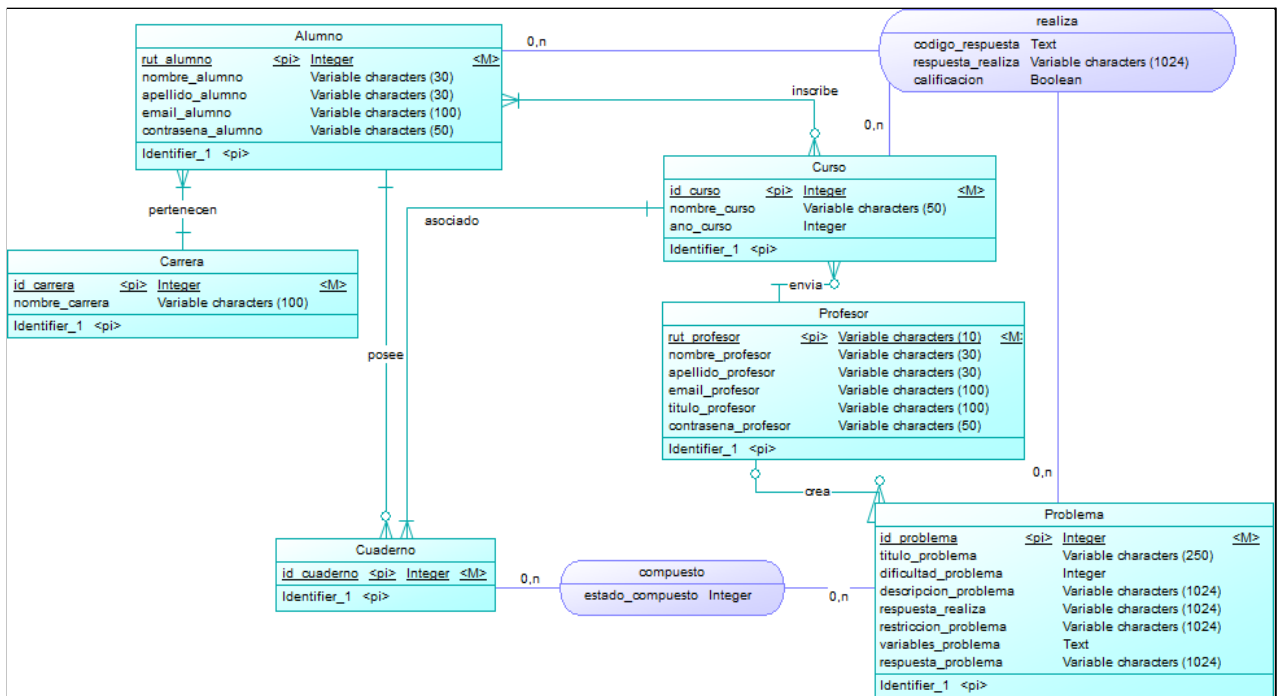


Figura 6.4.1: Modelo Entidad Relación

7 DISEÑO

7.1 Diseño Físico de la Base de datos

Este modelo físico representa la forma en que se administra la información del sistema de ejercitación en línea. La estructura de la base de datos está constituida por las siguientes tablas las cuales hacen que la base de datos creada sea consistente con el modelo entidad-relación presentado en el capítulo anterior en el apartado 6.4.

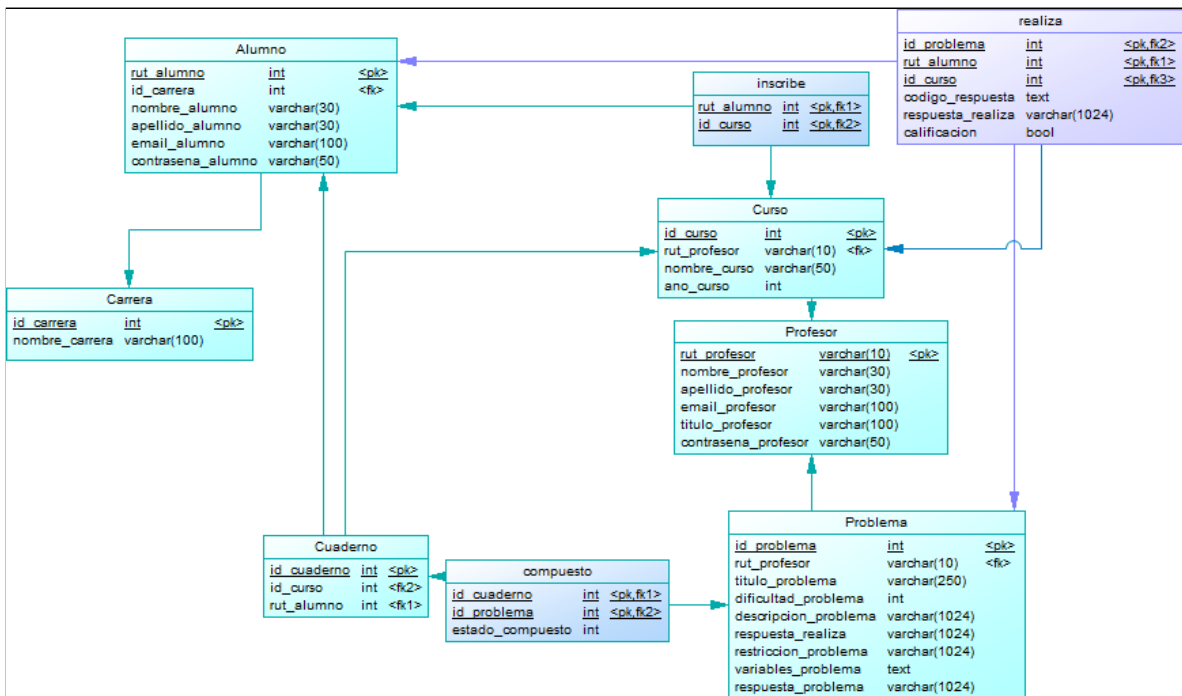


Figura 7.1.1: Diagrama modelo físico de la base de datos.

7.2 Diseño de arquitectura funcional

En el siguiente diagrama se representa el diseño de la arquitectura funcional del sistema a grandes rasgos.

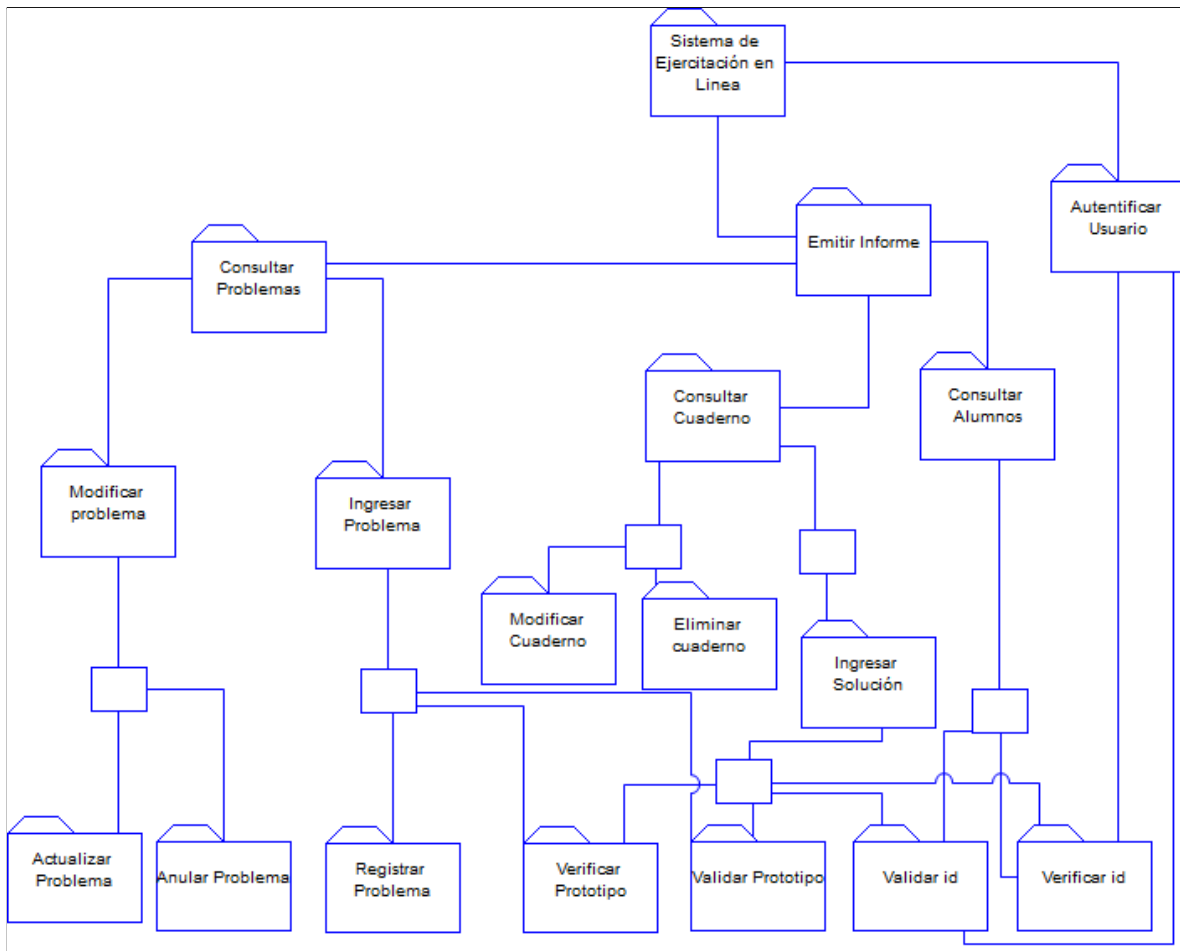


Figura 7.2.1: Árbol de descomposición del sistema.

7.3 Diseño interfaz y navegación

El diseño de la interfaz de usuario considera un diseño estándar que será respetado en todas las pantallas. Como se describe a continuación según los números del diagrama que consta de 5 secciones principales.

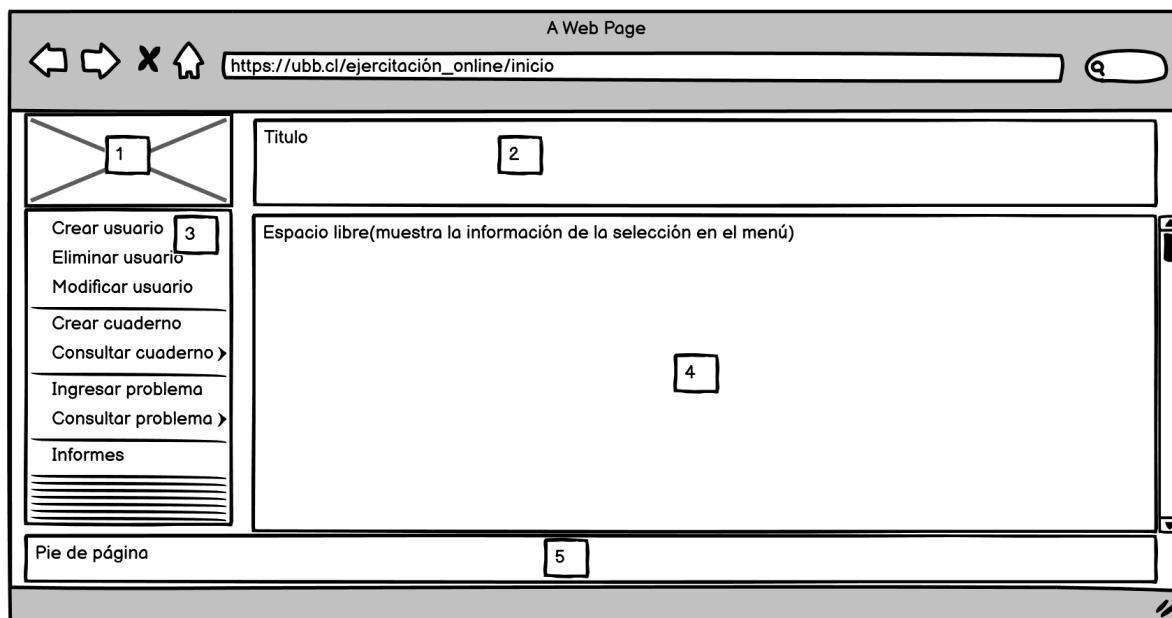


Figura 7.3.1: Diseño de interfaz general

1. Logo: Será utilizado en todas las pantallas del sitio, en él se encontrará el logo de la universidad.
2. Cabecera: Es utilizada en todas las páginas del sitio, con un fondo en degradado de color con tonalidad azul o verde, en ella se encuentra el título “Cuaderno de ejercitacion en linea”, con una imagen animada que haga referencia.
3. Menú: El menú se encuentra al lado izquierdo de la pantalla, el contiene las opciones disponibles para el usuario, cada una de estas opciones estarán referenciadas a un módulo.
4. Espacio Libre: En este espacio se muestra la opción seleccionada por el usuario, en el aparecerán todos los tópicos y estructuras de diseños para ese módulo por ejemplo,

listas, cuadernos y botones para las distintas funcionalidades que se le sean permitidas al usuario.

5. Pie de página: Se muestra información relevante sobre la plataforma como contactos, direcciones de ayuda y la condición para poder utilizar el software.

- Jerarquía de menú

La jerarquía de menú representa las conexiones y agrupaciones de las opciones de menú disponibles.

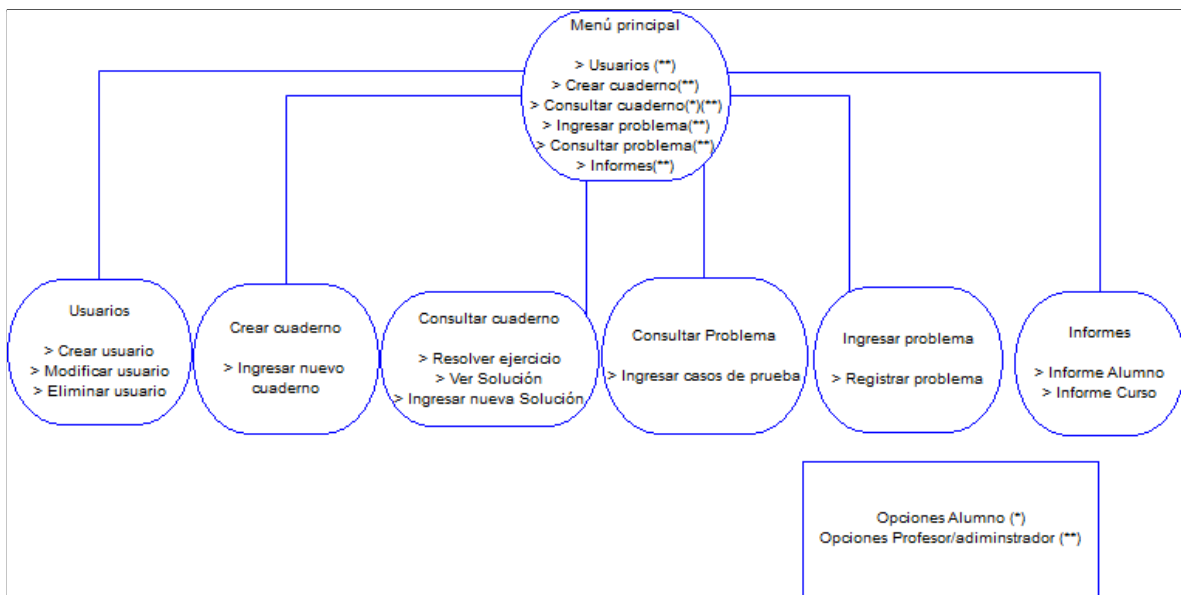


Figura 7.3.2 : Jerarquía de menú.

- Esquema de navegación

En el esquema de navegación se representan las opciones que tiene el usuario para navegar y/o recorrer dentro de las distintas opciones que el sistema ofrece, en este caso existen dos tipos de esquema de navegación, ya que los privilegios de los usuarios son distintos, así mismo se tomó a los profesores como disponibles para ejercer el cargo de administrador dentro el sistema.

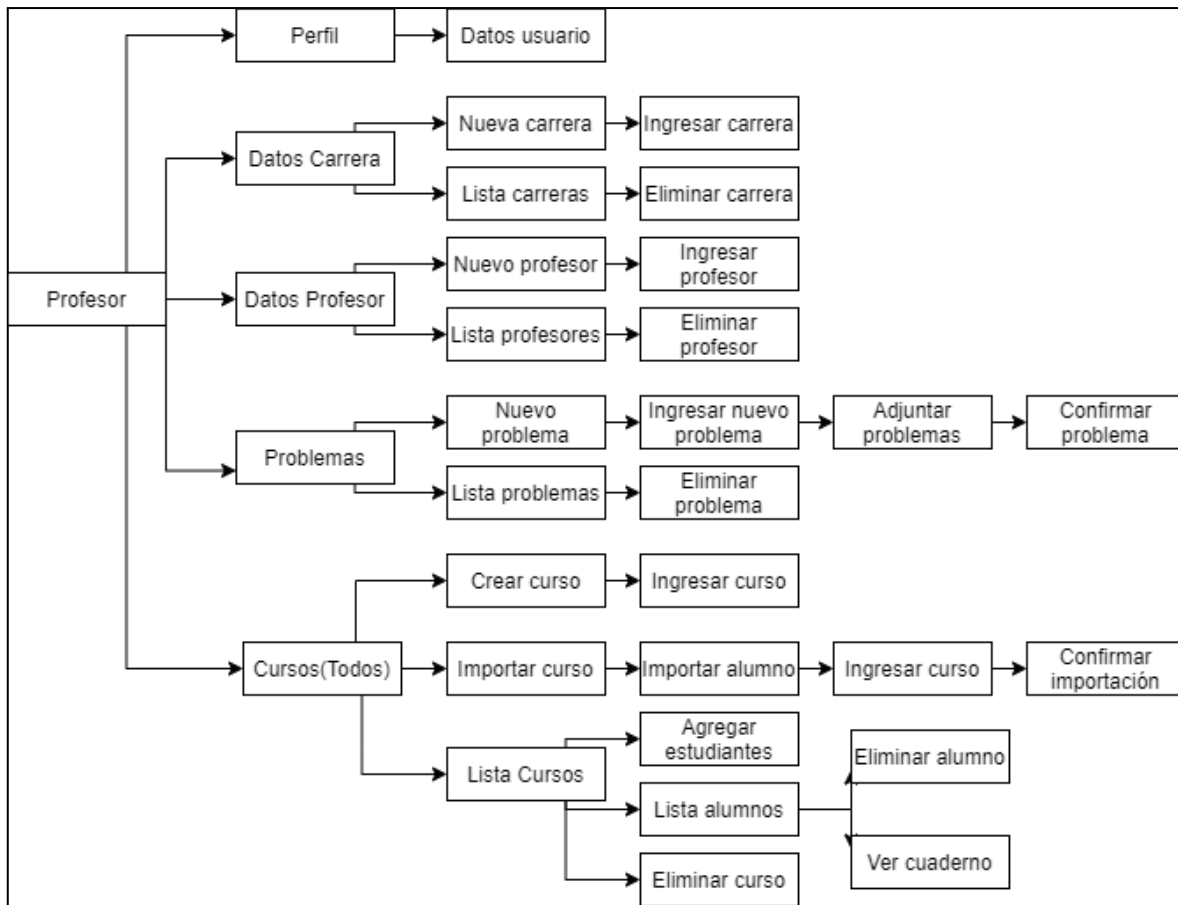


Figura 7.3.3: Esquema de navegación administrador.

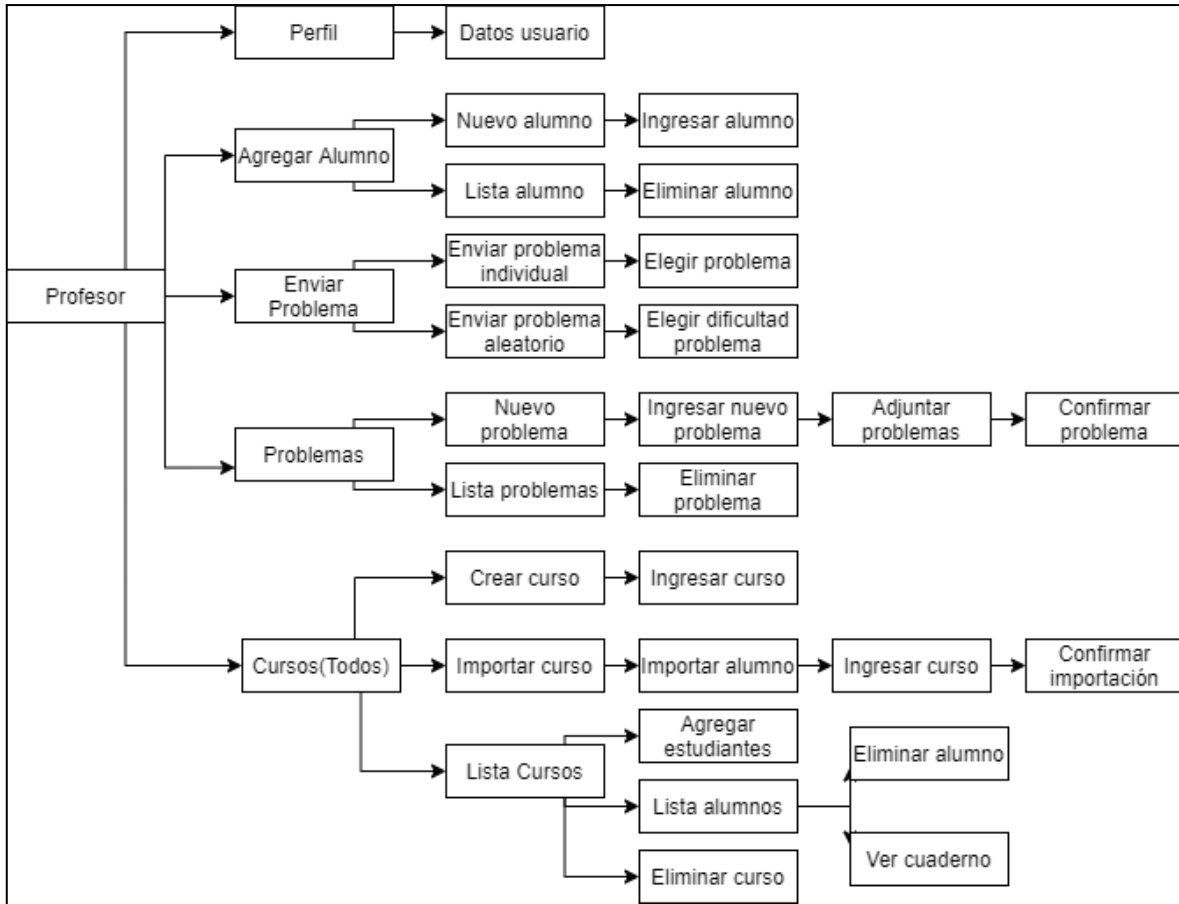


Figura 7.3.4: Esquema de navegación profesor.

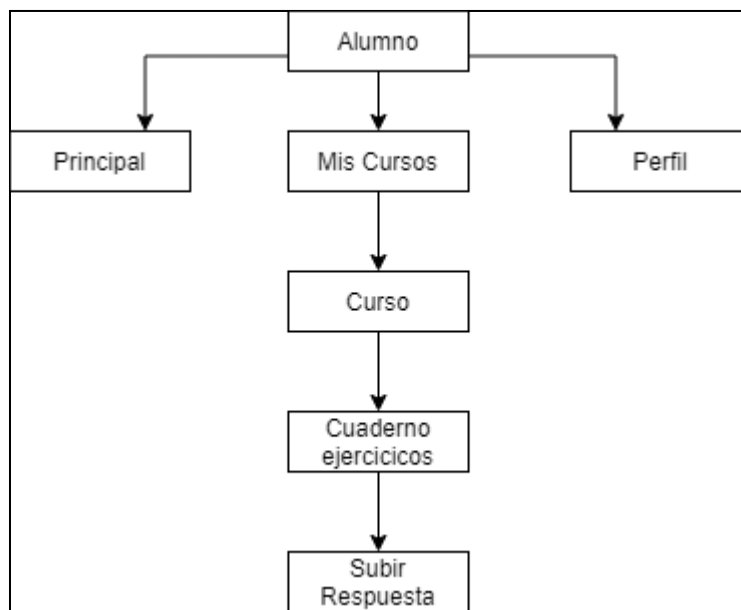


Figura 7.3.5: Esquema de navegación alumno.

7.4 Especificación de módulos

Los módulos de programa creados para esta aplicación se describen a continuación:

N° Módulo: 01		Nombre Módulo: Validar prototipo	
Parámetros de entrada		Parámetros de Salida	
Nombre:	Tipo de dato:	Nombre:	Tipo de dato:
prototipo_funcion	Variable carácter	Flag	Booleano

N° Módulo: 02		Nombre Módulo: Verificar prototipo	
Parámetros de entrada		Parámetros de Salida	
Nombre:	Tipo de dato:	Nombre:	Tipo de dato:
prototipo_funcion	Variable carácter	Flag	Booleano

N° Módulo: 03		Nombre Módulo: Registrar Problema	
Parámetros de entrada		Parámetros de Salida	
Nombre:	Tipo de dato:	Nombre:	Tipo de dato:
Tipo_ejercicio	Variable carácter	indicador registro	Booleano
enunciado	Variable carácter		
descripción	Variable carácter		
prototipo_funcion	Variable carácter		

N° Módulo: 04		Nombre Módulo: Actualizar Problema	
Parámetros de entrada		Parámetros de Salida	
Nombre:	Tipo de dato:	Nombre:	Tipo de dato:
Tipo_ejercicio	Variable carácter	Tipo_ejercicio	Variable carácter

enunciado	Variable carácter	enunciado	Variable carácter
descripción	Variable carácter	descripción	Variable carácter
prototipo_funcion	Variable carácter	prototipo_funcion	Variable carácter

N° Módulo: 05		Nombre Módulo: Eliminar Problema	
Parámetros de entrada		Parámetros de Salida	
Nombre:	Tipo de dato:	Nombre:	Tipo de dato:
id_problema	Variable carácter	indicador borrado	Booleano

N° Módulo: 06		Nombre Módulo: validar usuario	
Parámetros de entrada		Parámetros de Salida	
Nombre:	Tipo de dato:	Nombre:	Tipo de dato:
id_usuario	Variable carácter	Flag	Booleano
contraseña	Variable carácter		

N° Módulo: 07		Nombre Módulo: verificar usuario	
Parámetros de entrada		Parámetros de Salida	
Nombre:	Tipo de dato:	Nombre:	Tipo de dato:
id_usuario	Variable carácter	Flag	Booleano
contraseña	Variable carácter		

N° Módulo:08		Nombre Módulo: Crear cuaderno	
Parámetros de entrada		Parámetros de Salida	
Nombre:	Tipo de dato:	Nombre:	Tipo de dato:
id_alumno	Variable carácter	Flag	Booleano
c_descripcion	Variable carácter		

N° Módulo: 09		Nombre Módulo: modificar cuaderno	
Parámetros de entrada		Parámetros de Salida	
Nombre:	Tipo de dato:	Nombre:	Tipo de dato:
c_descripcion	Variable carácter	c_descripcion	Variable carácter
id_alumno	Variable carácter	id_alumno	Variable carácter

N° Módulo: 10		Nombre Módulo: Eliminar cuaderno	
Parámetros de entrada		Parámetros de Salida	
Nombre:	Tipo de dato:	Nombre:	Tipo de dato:
id_cuaderno	Variable carácter	indicador de borrado	Booleano

8 PRUEBAS

8.1 Elementos de prueba

Uno de los principales módulos que se van a probar es el de la conexión tanto a la base de datos con emisión y recepción de datos al sistema y la extracción de datos de usuarios para los distintos tipos de perfiles.

Las pruebas a realizar son de caja negra y son llevadas a cabo sobre la interfaz del software que es completamente indiferente al comportamiento interno y a la estructura el programa, con esto se pretende demostrar que las funcionalidades del software se encuentran operativas, los elementos de entrada sean verificados, adecuados y que se produzca una salida correcta de los datos.

Al momento de realizar las pruebas se determinaron algunos módulos internos que son :

- Problemas, se realizan pruebas de ingreso, de consultas y de eliminación de los problemas en el sistema.
- Casos de prueba, están ligados a los problemas, por lo que se debe probar la conexión entre ambas entidades.
- Respuestas alumnos, la cual está ligada a los problemas y cuadernos, por lo tanto se debe probar la conexión entre varias entidades.
- Cuaderno, donde se realizan pruebas de ingreso de problemas y solución de ejercicios tanto de manera individual cómo aleatoria.
- Ingresos de datos, donde se realizan pruebas de registro de alumnos, profesores, carreras, cursos, con sus respectivas consultas de datos y eliminaciones del sistema.

8.2 Especificación de las pruebas

Características a probar	Nivel de prueba	Objetivo de la Prueba	Enfoque para la definición de casos de prueba	Técnicas para la definición de casos de prueba	Actividades de prueba	Criterios de cumplimiento
Funcionalidad	sistema	Probar conexión entre el módulo problema	Caja negra	Particiones	1.- Ingresar al módulo problema. 2.-Acceder al módulo ingresar problema. 3.- Ingresar un nuevo problema. 4.- Ingresar al módulo consultar problemas. 5.- Eliminar problema ingresado	Verificar el correcto funcionamiento de las conexiones y funciones
Funcionalidad	Sistema	Probar conexión al módulo cuaderno	Caja negra	Particiones	1.- ingresar al módulo cuaderno. 2.- Ingresar a resolver un ejercicio.	Verificar el correcto funcionamiento de las

					<p>3.- Probar el ingreso de una solución al cuaderno.</p> <p>4.- Ingresar a ver la lista de ejercicios, para constatar los resueltos.</p>	<p>conexiones.</p>
Funcionalidad	Sistema	Probar conexión entre el módulo respuestas alumnos	Caja negra	Particiones	<p>1.- Ingresar al módulo enviar problemas.</p> <p>2.- Acceder al módulo donde se encuentra ingresar problema.</p> <p>3.- Ingresar al módulo subir respuesta.</p> <p>4.- Probar la función modificar respuesta</p>	<p>Verificar el correcto funcionamiento de las conexiones.</p>
Funcionalidad	sistema	Probar conexión entre casos de pruebas	Caja negra	Particiones	<p>1.- Ingresar al módulo de problemas.</p> <p>2.- Acceder al módulo ingresar problemas.</p>	<p>Verificar el correcto funcionamiento de las conexiones</p>

					3.- Ingresar diferentes casos de pruebas de diferentes ejercicios	es
Funcionalidad	sistema	Probar los diferentes módulos de ingreso de datos	Caja negra	Particiones	1.- Ingresar a los diferentes módulos de registro de datos(alumnos, profesores, carrera, cursos). 2.- Acceder a los diferentes módulos. 3.- Registrar diferentes datos. 4.- Probar la eliminación de datos.	Verificar el correcto funcionamiento de las conexiones.

Tabla 6: Especificación de las pruebas.

8.3 Responsables de las pruebas

Las personas encargadas y responsables de realizar las pruebas en el sistema de forma correcta son los desarrolladores del software, para probar las funcionalidades y las conexiones a los distintos módulos con los que interactúa el sistema entre sí.

8.4 Calendario de pruebas

Las pruebas realizadas se hicieron a medida que se fue terminando la implementación de los módulos y así verificar que el sistema está trabajando de manera correcta oportunamente.

8.5 Detalle de las pruebas

Los detalles de las pruebas se encuentran en el anexo 17 del presente documento.

8.6 Conclusiones de Prueba

Las conexiones del sistema se consideran bien realizadas, ya que interactúan de forma correcta con la base de datos al ingresar, consultar y eliminar datos del sistema, siendo estas probadas de forma local y mostrar la respuesta esperada.

9 CONCLUSIONES

Lo expuesto a lo largo del desarrollo de este trabajo nos permite arribar a una solución web capaz de satisfacer los objetivos planteados en los capítulos iniciales, pudiendo esta ser un gran aliado dentro de las asignaturas al momento de practicar lo aprendido de clases teóricas en las clases prácticas para el alumno y para el profesor una plataforma capaz de poder facilitar la ejercitación de los alumnos fuera de clases.

Durante el estudio, diseño y los distintos capítulos que componen este informe se seleccionaron herramientas modernas que se sitúan al nivel óptimo de las últimas tecnologías con la consideración de que estas se adapten a las herramientas con las que cuenta la universidad al día de hoy, para lograr una compatibilidad total al momento de su implementación. En la elección del lenguaje y herramientas se consideró la oportunidad de que hubiese a futuro una expansión en el software o posibles mejoras sin afectar el rendimiento inicial.

Para finalizar cabe mencionar que la problemática del trabajo generó un gran interés en el equipo al poder brindar apoyo a los nuevos estudiantes que se integren a la carrera, junto con la experiencia adquirida a lo largo del proyecto que será de gran ayuda para la primera inserción como profesionales al mundo laboral.

9.1 Trabajos futuros

Durante el desarrollo de este proyecto, se han definido, obtenido resultados y experiencias que dan espacio a un conjunto de ideas, trabajos y tareas que son necesarias abordar en trabajos futuros.

- Implementación del proyecto en el servidor de la Universidad para su futuro uso
- Implementar la funcionalidad de modificación de datos mencionados en secciones anteriores, este modelo incorpora los siguientes módulos:
 - Modificar Datos Alumno.
 - Modificar Datos Profesor.
 - Modificar Datos Problemas.
 - Modificar Datos Cursos.
 - Modificar Datos Cuadernos.

10 BIBLIOGRAFÍA

- Alaín, S. B. E. (2015). Sistema Interactivo Basado En Un Intérprete De Algoritmos Para Mejorar El Método De Aprendizaje De Los Alumnos Del Curso Fundamentos De Programación. 82. http://tesis.usat.edu.pe/bitstream/usat/643/1/TL_Barturen_Guevara_JheniferMishel.pdf
- Bartó, C. A., & Díaz, L. C. (2012). PROYECTO : SISTEMAS INTELIGENTES APLICADOS A LA ENSEÑANZA DE LA PROGRAMACIÓN EN INGENIERÍA Resumen Contexto Introducción. XIV Workshop de Investigadores En Ciencias de La Computación, 1051–1055.
- Grado, T. D. F. De, Informática, F. De, & Complutense, U. (2013). Juez Automático para la Evaluación de Problemas de Programación en los Primeros Cursos de Estudios de Informática.
- Programación Estructurada. (n.d.). *Modelo Incremental o Evolutivo*. Programación Estructurada. Retrieved Diciembre 12, 2020, from <https://sites.google.com/site/programacion1electronica/metodologias-de-desarrollo-de-software/modelo-incremental-o-evolutivo>
- Soler Pellicer, Y., Cárdenas Zea, M., Aguirre Pérez, R., Castro Blanco, Y., & Lezcano Brito, M. G. (2020). Visualización Dinámica, Una Opción Para La Enseñanza-Aprendizaje De La Programación De Computadoras. *Holos*, 2, 1–20. <https://doi.org/10.15628/holos.2020.4241>
- Universidad del Bío-Bío. (n.d.). Universidad del Bío-Bío. Retrieved Diciembre 12, 2020, from https://www.ubiobio.cl/w/#Vision_y_Mision

11 ANEXO: PLANIFICACIÓN INICIAL DEL PROYECTO

Actividad	Rango de fechas (dd/mm/aa)
1 Etapa de Planificación	(26-10-2020)~(16-11-2020)
1.1 Planificación de Gantt	(26-10-2020)~(27-10-2020)
1.2 Estudio del Arte	(28-10-2020)~(9-11-2020)
1.2.1 Buscar soluciones y definir solución	(28-11-2020)~(9-11-2020)
1.3 Definir proyecto	(10-11-2020)~(13-11-2020)
1.5 Mostrar Avance	(16-11-2020)~(16-11-2020)
2 Etapa de Análisis	(17-11-2020)~(22-12-2020)
2.1 Toma de requisitos	(17-11-2020)~(25-11-2020)
2.2 Definir software	(26-11-2020)~(30-11-2020)
2.3 Requisitos específicos	(1-12-2020)~(2-12-2020)
2.4 Realización diagrama de casos de uso	(3-12-2020)~(18-12-2020)
2.5 Modelo de datos	(21-12-2020)~(22-12-2020)
2.6 Mostrar Avance	(23-12-2020)~(23-12-2020)
3 Etapa de Diseño	(24-12-2020)~(15-1-2021)
3.1 Diseño de la BDD	(24-12-2020)~(28-12-2020)
3.2 Elaborar arquitectura	(29-12-2020)~(4-1-2021)
3.3 Diseño de interfaz de usuarios	(5-1-2021)~(11-1-2021)

3.4 Diseño de pruebas de sistema	(12-1-2021)~(14-11-2021)
3.5 Mostrar Avance	(15-12-2020)~(15-12-2020)
4 Etapa de Implementación	(19-2-2021)~(21-8-2021)
4.1 Implementar BDD	(19-2-2021)~(22-2-2021)
4.2 Primera Iteración	(4-4-2021)~(23-5-2021)
4.2.1 Desarrollo Interfaz	(4-4-2021)~(2-5-2021)
4.2.2 Revisión de Versión	(5-5-2021)~(8-5-2021)
4.2.3 Correcciones de versión	(9-5-2021)~(14-5-2021)
4.2.4 Implementar pruebas	(15-5-2021)~(22-5-2021)
4.2.5 Mostrar Avance	(23-5-2021)~(23-5-2021)
4.3 Segunda Iteración	(26-5-2021)~(4-7-2021)
4.3.1 Desarrollo	(26-5-2021)~(21-6-2021)
4.3.2 Segunda Revisión de versión	(24-6-2021)~(26-6-2021)
4.3.3 Correcciones de Revisión	(27-6-2021)~(31-6-2021)
4.3.4 Implementar pruebas	(1-7-2021)~(3-7-2021)
4.3.5 Mostrar Avance	(4-7-2021)~(4-7-2021)
4.4 Tercera Iteración	(7-7-2021)~(19-8-2021)
4.4.1 Desarrollo	(7-7-2021)~(5-8-2021)
4.4.2 Tercera Revisión de versión	(6-8-2021)~(8-8-2021)
4.4.3 Correcciones de Revisión	(9-8-2021)~(13-8-2021)
4.4.4 Implementar pruebas	(13-8-2021)~(15-8-2021)

4.4.5 Mostrar Avance	(16-8-2021)~(16-8-2021)
4.5 Desarrollo de documentación	(16-8-2021)~(21-8-2021)
4.5.1 Manuales de usuario	(16-8-2021)~(20-8-2021)
4.5.3 Otros	(20-8-2021)~(21-8-2021)
4.6 Mostrar Avance final	(21-8-2021)~(21-8-2021)
5 Entrega de Proyecto Final	(23-8-2021)
Entrega del Proyecto (Anteproyecto)	(22-02-2021)

12 ANEXO: RESULTADOS DE ITERACIONES EN EL DESARROLLO

➤ 1º iteración

Durante el desarrollo de la 1era iteración se lleva a cabo la instalación e implementación de la base de datos en el espacio otorgado por la universidad para poder crear las tablas y alojar los datos de prueba de la base de datos.

Se visualiza de la siguiente manera:

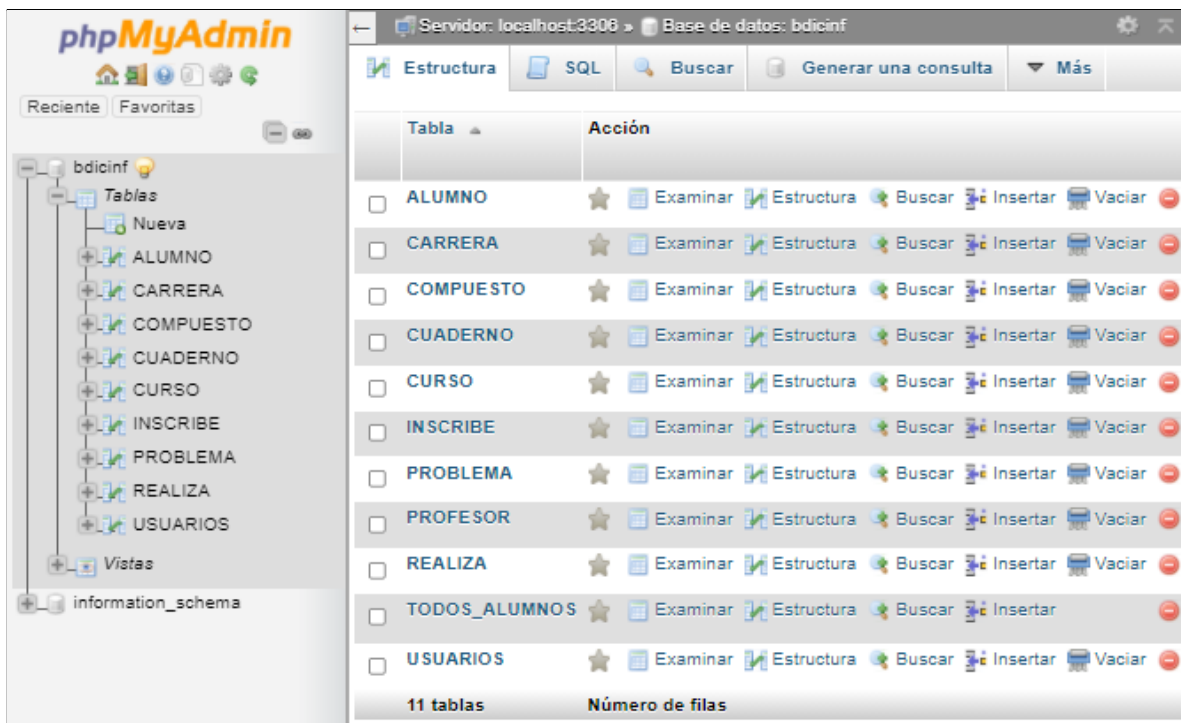


Figura 12.1: Base de Datos MYSQL

Una vez con la base de datos funcionando de manera correcta, se procede a crear cuentas de usuarios y otros datos para hacer consultas de prueba.

➤ 2º Iteración:

Se inicia con la creación de un sistema web desarrollado bajo el lenguaje de programación React, el que se conecta con una base de datos Mysql usando Node js, siguiendo la separación de código en componentes para mantener el mayor orden posible en él desarrollo, se procede a crear un login para los inicio de sesión y el perfil de profesor inicialmente para poder ir probando las conexiones del back-end a la base de datos.

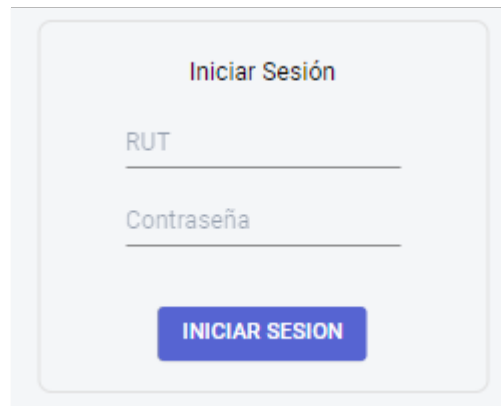


Figura 12.2: Inicio de sesión

Una vez creado el inicio de sesión se procedió a desarrollar el perfil de profesor, con todas sus opciones dentro de la plataforma.

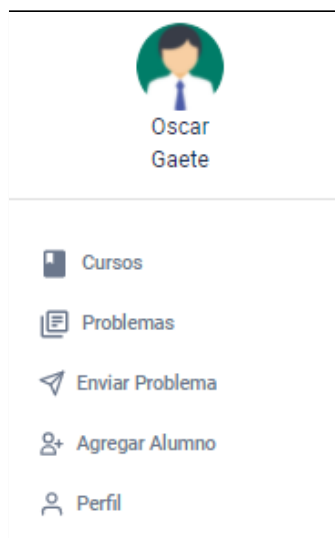


Figura 12.3: Menú de Opciones Profesor.

➤ 3° Iteración:

Se inicia con la creación del perfil de alumno de la aplicación, con sus respectivas opciones y menos privilegios, se crean los componentes y vistas para los apartados de ver cuaderno y para subir respuestas.

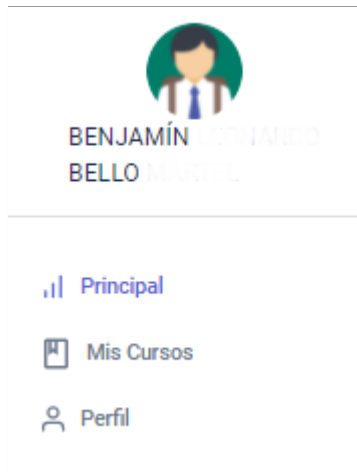


Figura 12.4: Menú Opciones Alumno

En la opción de Mis Cursos el cliente sugirió que se mostrara los datos tanto de ejercicios, como los del profesor para que el alumno tuviera a mano la información para comunicarse en caso de dudas, viéndose de la siguiente manera:

VOLVER

MIS CURSOS

Curso: Introducción a la Ingeniería(2021)

Profesor: Rodolfo Cuevas Pantoja
Email: rcuevaspantoja@gmail.com

- ✓ Ejercicios Correctos: 0
- ✗ Ejercicios Incorrectos: 0
- Ejercicios Sin Responder: 3

VER CURSO

Figura 12.5: Mis Cursos Alumno.

Posteriormente para la subida de respuestas el cliente pensó en crearla de tal manera que el alumno tuviera la opción de ver en pantalla si el resultado de su código compilado para ver si su respuesta era la esperada, a través de una consola en pantalla. El alumno debe subir un archivo donde se encuentre su código respuesta de acuerdo al ejercicio. Mientras espera que se ejecute su tarea, visualiza una pantalla como esta:

The screenshot shows a web interface for submitting an answer to a programming question. At the top left is a blue button labeled 'VOLVER'. The question title is 'Pregunta 36: Número entre 100~200'. The question text asks for an algorithm to check if a real number is between 100 and 200. Restrictions include using 'if-then-else' structures. The difficulty is indicated by five stars, with three yellow and two grey. Below is a section for attaching a response, with a button 'Seleccionar archivo' and the text 'Ningún archivo seleccionado'. A blue button 'COMPILAR' is present. A 'Consola:' label is followed by a blacked-out area. At the bottom center is a grey button 'CONTINUAR'.

Figura 12.6: Subida Respuesta Alumno.

4° Iteración:

Se procede con la mejora el perfil del administrador por una solicitud hecha por el cliente en el cuál nos solicita qué administrador tenga los permisos sobre todos los datos de profesores y cursos para sus modificaciones, también se solicitó la creación de un módulo para qué el profesor importara cursos completos desde un archivo xlsx con el formato intranet.

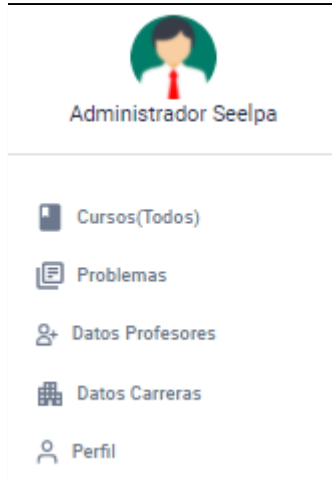


Figura 12.7: Menú de Opciones Administrador

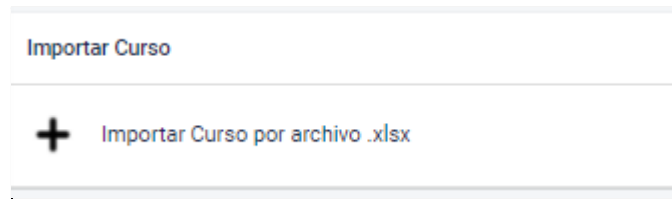


Figura 12.8: Módulo Importar Curso Archivo.xlsx.

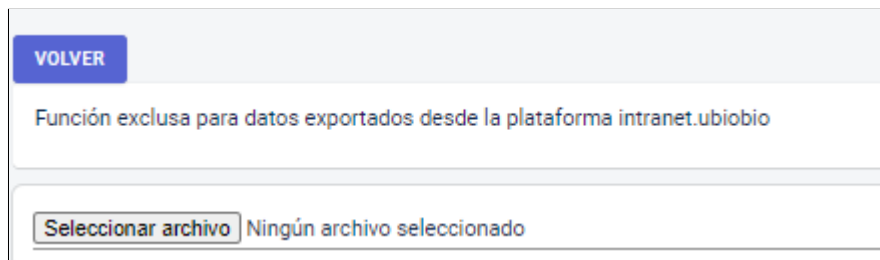


Figura 12.9: Espacio para Importar Curso Perfil Profesor.

Con esto realizado se dio por casi finalizada la cuarta iteración, dado que ya se habían cumplido en gran parte con las solicitudes hechas por el cliente, obteniendo así un visto bueno de su parte hacia el producto.

13 ANEXO: MANUAL DEL USUARIO

Para una correcta utilización del sistema SEELPA es necesaria la documentación para un uso correcto del sistema y sus partes. El sistema cuenta con tres distintos tipos de usuarios dependiendo del tipo de cuenta a la cual se obtiene acceso mediante un proceso de autenticación, las cuentas tienen diferentes privilegios y opciones de visualización, de esta manera se dividirá el manual de acuerdo a estos roles. Los roles existentes en el sistema son: Administrador, Profesor y Alumno.

13.1 Administrador

Al autenticarse como una sesión de tipo Administrador, se obtendrá acceso a un menú similar a la de la siguiente imagen. En este menú se muestra el nombre de usuario con el cual se inició sesión y las opciones que tiene un rol de administrador que consiste en Cursos(todos), Problemas, Datos Profesor, Datos Carreras y Perfil. Como lo indica el nombre del rol, está creado para poder dar soporte al sistema, crear cuentas para los profesores, registrar nuevas carreras para que más alumnos puedan hacer uso de la plataforma, crear cursos a nombre de profesores que tengan problemas creando los suyos propios y poder eliminar cursos si a un docente se le olvida hacerlo a final de semestre.

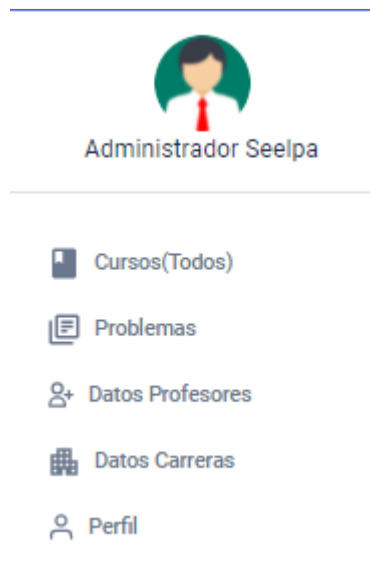


Figura 13.1.1: Menú de Administrador.

a) Cursos(Todos)

En la opción del menú de Cursos(Todos) aparecerán 2 distintas formas de crear cursos y en la tabla que continúa se mostrarán la totalidad de cursos creados, que a diferencia de un rol de profesor, que solo le enseñaría los cursos creados bajo su rut, este muestra la totalidad de cursos para permitirle al administrador poder hacer modificaciones en cursos que no ha creado él como agregar alumnos o eliminarlos en caso de ser necesario y en caso de ser necesario crear y eliminar los cursos.

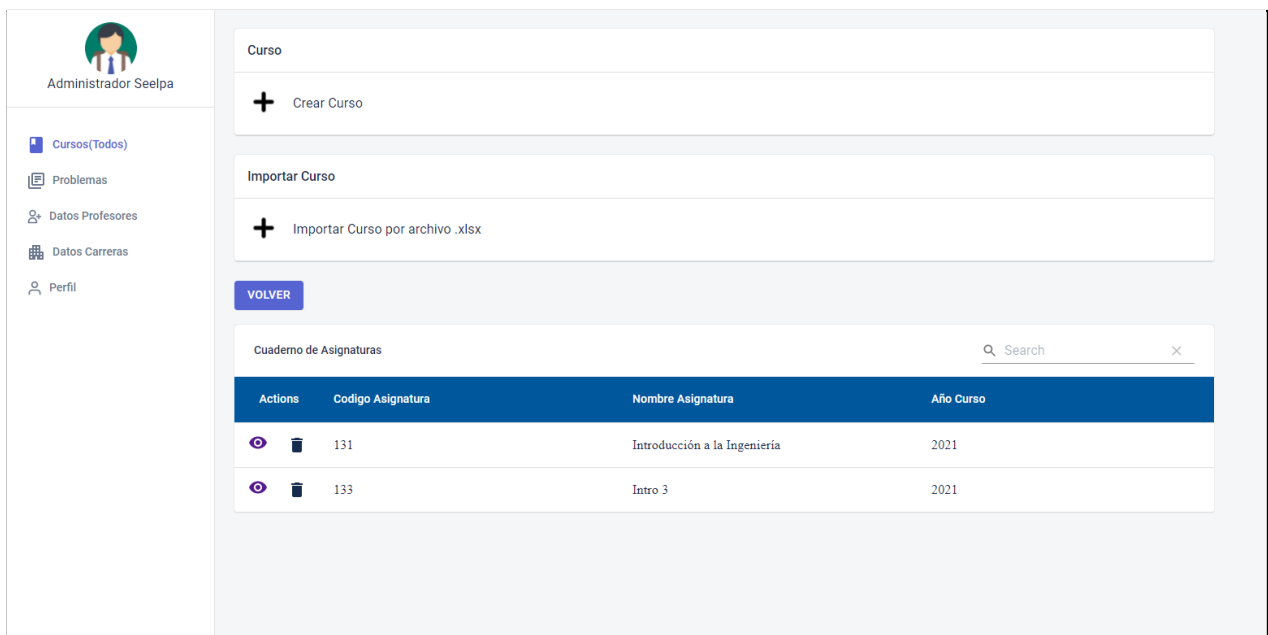
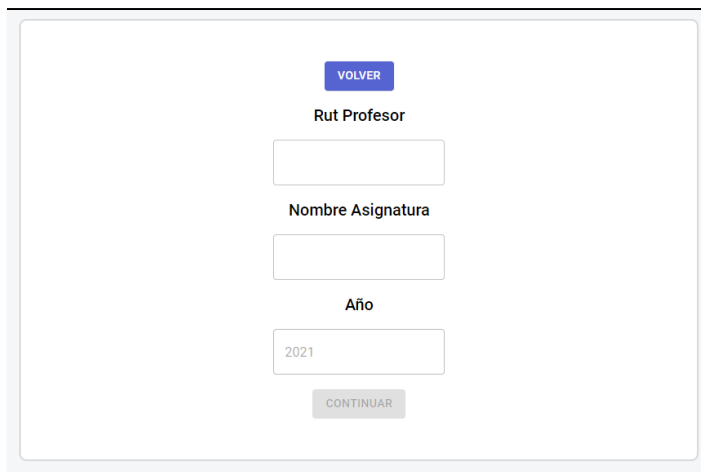


Figura 13.1.2: Opciones del menú Cursos(Todos)

Tras hacer click sobre la cruz en Crear curso, se desplegará un menú como el siguiente el cual permite crear un curso a nombre de un profesor completando el formulario.



Formulario para crear un curso administrador. El formulario contiene los siguientes elementos:

- Botón "VOLVER" (azul) en la parte superior.
- Etiqueta "Rut Profesor" con un campo de entrada de texto vacío.
- Etiqueta "Nombre Asignatura" con un campo de entrada de texto vacío.
- Etiqueta "Año" con un campo de entrada de texto que contiene el valor "2021".
- Botón "CONTINUAR" (gris) en la parte inferior.

Figura 13.1.3: Crear Curso Administrador

Tras hacer click sobre la cruz en Importar Curso por archivo .xlsx se accede al menú para subir el archivo con la lista de estudiantes (cabe aclarar que el software solo acepta los archivos exportados desde la plataforma intranet.ubiobio). Una vez seleccionado el archivo, este será leído y mostrará en formato de tarjetas cada dato de los estudiantes a manera de que el profesor pueda revisar que se va a realizar la importación de manera correcta. Al presionar continuar se mostrará el formulario anterior para la creación del curso al cual serán importados estos Alumnos. Finalmente, se mostrará una ventana a manera de confirmación de lo realizado anteriormente junto con el botón para empezar a crear las cuentas, cuadernos e inscripciones de los alumnos en el curso creado en el paso anterior.

VOLVER

Función exclusiva para datos exportados desde la plataforma intranet.ubiobio

Seleccionar archivo | No se eligió archivo

CONTINUAR

Figura 13.1.4: Importar Curso Administrador

VOLVER

Preparando a 27 Alumnos para ser importados...

Rut Profesor

Nombre Asignatura

Año

2021

CONTINUAR

Figura 13.1.5: Crear Curso Importado Administrador



Figura 13.1.6: Completar Importación Administrador

Tras hacer click encima del ojo en la tabla curso mostrará todos los datos del curso, junto con estadísticas grupales para poder ver el desempeño del curso y los alumnos que no han participado con la opción de poder eliminar un estudiante y la opción para poder revisar el cuaderno individual del alumno. En este apartado se mostrará de amarillo con un "Problema Pendiente" en caso de que el alumno no haya realizado aún la actividad, "Completado Erróneamente" si el alumno mandó de manera errónea el problema y un "Completado Exitosamente" si lo realizó de manera correcta el problema.

VOLVER AGREGAR ESTUDIANTES


ESTADÍSTICAS DEL CURSO

Total de Problemas Resueltos: 2 / 54
 Total de Problemas Correctos: 1 / 2
 Total de Problemas Incorrecto: 1 / 2
 Alumnos Que No Han Participado: 27 Ver Alumnos

Intro 3 Search X

Actions	Rut	Nombre	Carrera	Correo
 	18987529-0	Gerson Pereira Fontalba	(2927) INGENIERÍA CIVIL EN INFORMÁTICA	gerson.pereira1501@alumnos.ubiobio.cl
 		CLAUDIA	(2927) INGENIERÍA CIVIL EN INFORMÁTICA	@ALUMNOS.UBIOBIO.CL

Figura 13.1.7: Participantes de un curso Administrador

 Esta acción eliminará de manera permanente el cuaderno del alumno.

Desea eliminar a **Gerson Pereira Fontalba** del curso **Intro 3?**

CONFIRMAR
NO

Figura 13.1.8: Eliminar un Alumno Administrador

✓ Ejercicios Correctos: 0
✗ Ejercicios Incorrectos: 0
● Ejercicios Sin Responder: 2

Problema Pendiente

Problema 44: Resolver Mínimo Común divisor entre 2 números ingresados
 Restricción: limitarse al uso de if, else, do-while
 Descripción: mediante 2 datos de entrada calcular el MCD entre ambos

Problema Pendiente

Problema 42: Hola Mundo
 Restricción: sin restricciones
 Descripción: Programar un código que muestre en consola un "Hola Mundo es un programa sin entradas"

CERRAR

Figura 13.1.9: Cuaderno Alumno parte 1 Administrador

Completado Erróneamente

Problema 44: Resolver Mínimo Común divisor entre 2 números ingresados
 Restricción: limitarse al uso de if, else, do-while
 Descripción: mediante 2 datos de entrada calcular el MCD entre ambos

Respuesta:

```

1 #include <stdio.h>
2
3 int main(){
4
5     int a = 0;
6     int b = 0;
7     int c = 0;
8     int d = 0;
9
10    scanf("%d", &a);
11    scanf("%d", &b);
12    scanf("%d", &c);
13    scanf("%d", &d);
14    printf("A=%d B=%d C=%d D=%d", a,b,c,d);
15
16    return 0;
17
18 }
    
```

Completado Exitosamente

Problema 42: Hola Mundo
 Restricción: sin restricciones

CERRAR

Figura 13.1.10: Cuaderno Alumno parte 2 Administrador

b) Problemas

En la opción del menú Problemas se encontrará la opción para la creación de un problema al apretar la cruz en Agregar Problema junto con la “Lista de Ejercicios” que contempla todos los ejercicios disponibles para agregar a los cuadernos de los estudiantes, permitiendo ver la solución de este al presionar el símbolo con forma de ojo y al presionar la figura con forma de basurero saldrá la opción para confirmar y borrar el problema (esta opción debe ser usada sólo para cuando un problema esté mal subido a la plataforma, ya que lo eliminará de los cuadernos de los estudiantes).

Problemas

+ Agregar Problema

VOLVER

Lista de Ejercicios 🔍 Search ✕

Actions	Autor	Título	Dificultad	Restricción	Variables	Respuesta
<input type="radio"/> <input type="checkbox"/>	Carlos	Número entre 100-200	3	Uso de las estructuras de selección if-then-else.	150	Está dentro del intervalo
<input type="radio"/> <input type="checkbox"/>	Carlos	Si un numero es divisor del otro	4	Uso de las estructuras de selección if-then-else.	10 5	Es divisor 10 de 5
<input type="radio"/> <input type="checkbox"/>	Carlos	Función recursiva que regrese el máximo común divisor de X y de Y	5	Use recursividad para este problema, así como las estructuras de control selectivas y repetitivas.	100 50	El MCD de x=100 y y=50 es: 50
<input type="radio"/> <input type="checkbox"/>	Carlos	Test Variables de Entrada	2	sin restricciones	1 2 3 4	A=1 B=2 C=3 D=4
<input type="radio"/> <input type="checkbox"/>	Carlos	Hola Mundo	1	sin restricciones		Hola Mundo es un programa sin entradas

Figura 13.1.11: Menú Problemas Administrador


```

4 {
5 int Numero1=0, Numero2=0; //Variables
6
7 scanf("%i", &Numero1); //indica sin signo
8 scanf("%i", &Numero2); //indica sin signo
9 if (Numero1 > Numero2)
10 {
11 if (Numero1%Numero2 == 0)
12 {
13 printf("Es divisor %i de %i", Numero1,
14 Numero2);
15 }
16 else
17 printf("%i no es divisor de %i ", Numero2,
18 Numero1);
19 }
20 else
21 printf("No es divisor porque es mayor %i",
22 Numero2);
23 return 0;
24 }

```

CERRAR

Figura 13.1.12: Mostrar Respuesta Problema Administrador

Agregar Problema cuenta con un formulario el cual solicita Título, Descripción, Restricción, dificultad y variables de entrada en caso de ser necesarias. Siguiendo de una ventana donde se debe subir el archivo o código que es la solución al problema, se requerirá que el código compile, para lo cual se debe presionar el botón compilar lo cual ejecutará el código subido por el administrador para que el pueda corroborar que la respuesta es la esperada. Finalmente, se podrá ver una ventana donde se verá toda la información y solución del problema para terminar siendo subido a la piscina de problemas para que cualquier profesor pueda hacer uso de la pregunta.

VOLVER

Título del Problema

Descripción del Problema

Restricción

Dificultad del Problema

★★★★★
La dificultad es / 5

i

Variables de Entrada (No? Dejar vacío)

CONTINUAR

Figura 13.1.13: Agregar Problema Administrador

VOLVER

Título: Número entre 100 y 200
Descripción: ingresado un número ver si este esta entre 100 y 200
Restricción: Uso de las estructuras de selección if-then-else.
Dificultad: 3 / 5 ★
Variables: 150

SELECCIONAR ARCHIVO:

Seleccionar archivo Problema de Intervalos.c

COMPILAR

Consola:

Está dentro del intervalo

SUBIR PREGUNTAR

Figura 13.1.14: Procesar Pregunta Administrador

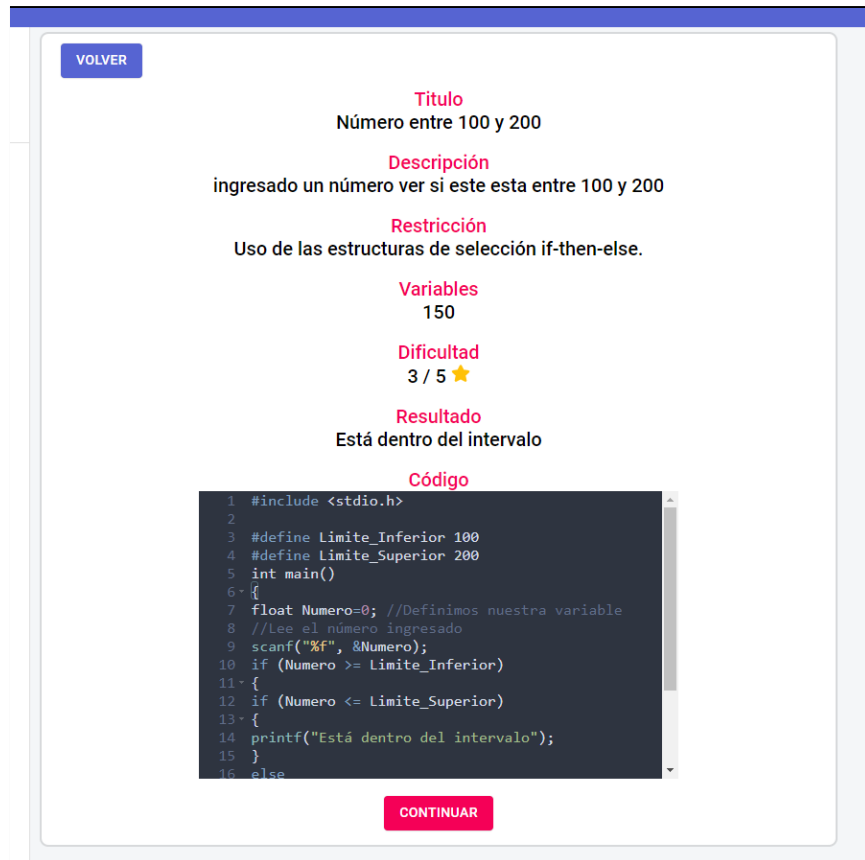


Figura 13.1.15: Confirmar Respuesta Administrador

c) Datos Profesor

En la opción del menú Datos Profesor se encuentra la opción para poder ingresar un usuario con rol de profesor y la lista de profesores con cuenta en la plataforma, de manera que el administrador pueda administrar las cuentas de los profesores que harán uso de la plataforma. Al hacer click en “Ingresar Profesor” se desplegará el formulario para completar los datos de un profesor.

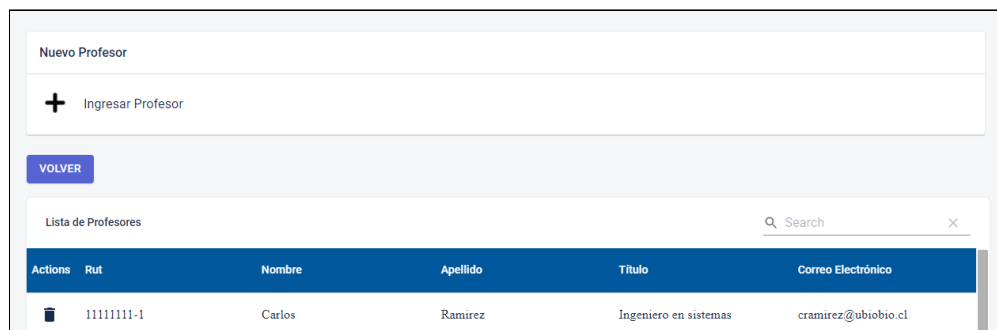


Figura 13.1.16: Menú Profesores Administrador

VOLVER

i
Rut Profesor

Nombre Profesor

Apellido Profesor

Correo Electrónico

Título Profesor

Contraseña

CONTINUAR

Figura 13.1.17: Agregar Profesor Administrador

d) Datos Carrera

En la opción del menú Datos Carrera se encuentra todo lo necesario para que el administrador pueda crear las carreras que harán uso del software y actualizar en caso de ser necesario los códigos de carrera, para que esté actualizado a la plataforma de Intranet Ubiobio y no tenga problemas aceptando los archivos exportados de la misma plataforma. Al hacer click en "Ingresar Nueva Carrera" se desplegará el formulario para completar los datos de carrera.

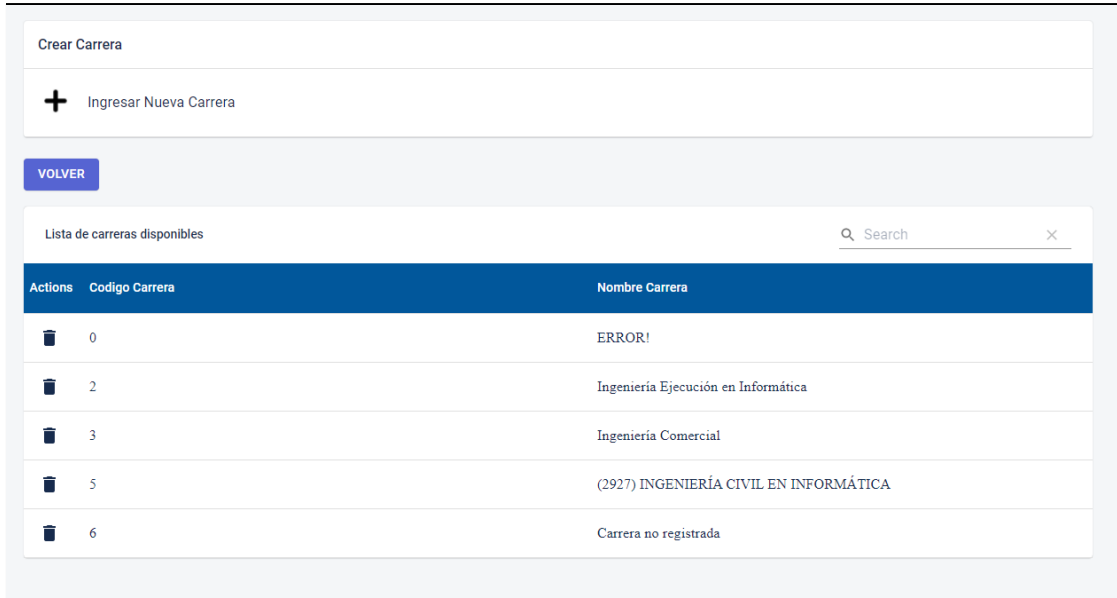


Figura 13.1.18: Menú Carrera Administrador

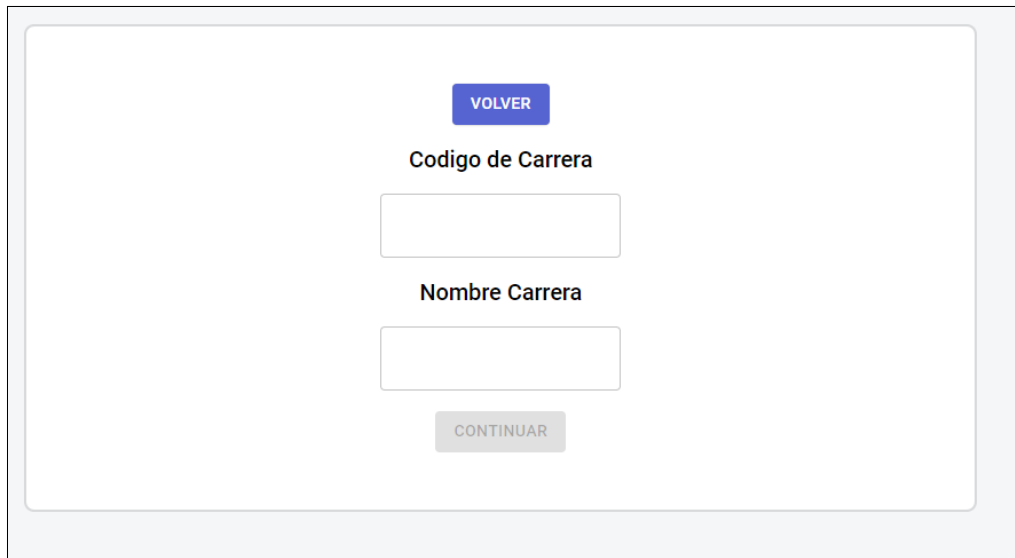


Figura 13.1.19: Agregar Carrera Administrador

e) En la opción del menú Perfil se encontrará los datos de la cuenta con lo cual se inició sesión de manera que el administrador pueda ver si foto de perfil correspondiente a administrador, rango, correo, título, etc.

13.2 Profesor

Al autenticarse como una sesión de tipo Profesor, se obtendrá acceso a un menú similar al de la siguiente imagen. En este menú se muestra el nombre del usuario con el cual se inició sesión y las opciones que tiene un rol de profesor que consiste en Cursos, Problemas, Enviar Problema, Agregar Alumno y Perfil. Como lo indica el nombre del rol, está creado para poder dar funcionamiento al sistema, crear cuentas para los alumnos si es él caso, registrar nuevos problemas para que exista más variedad en la plataforma, crear cursos que tengan problemas creando los suyos propios y poder eliminar sus cursos y alumnos el curso que ya no lo rindan.

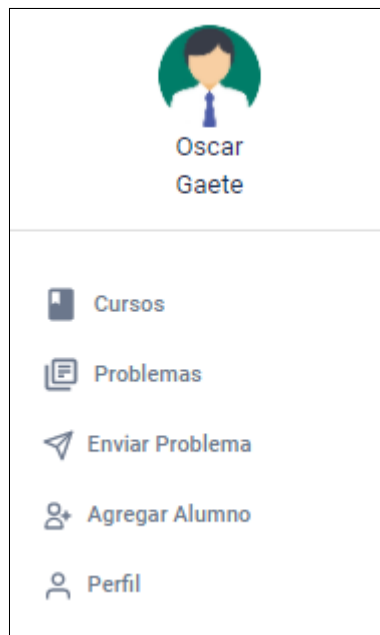


Figura 13.2.1 Opciones Menú Profesor

a) Cursos

En la opción del menú de Cursos aparecerán 2 distintas formas de crear cursos de forma manual y por importación, y en la tabla que continúa se mostrarán la totalidad de cursos creados asignados al profesor.

Al completar el formulario con los datos, el rut del profesor se toma de manera automática desde las credenciales de inicio de sesión, luego se habilitará el botón continuar y al hacer click se registrara el curso.

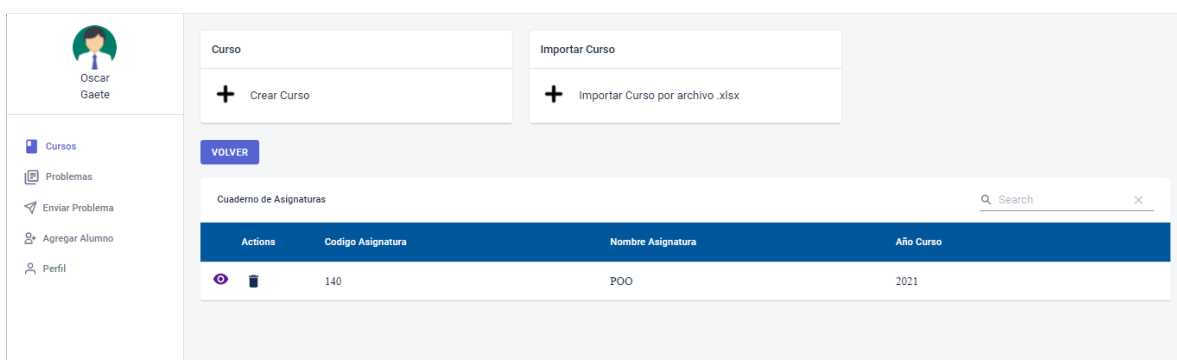


Figura 13.2.2 Opciones de Cursos

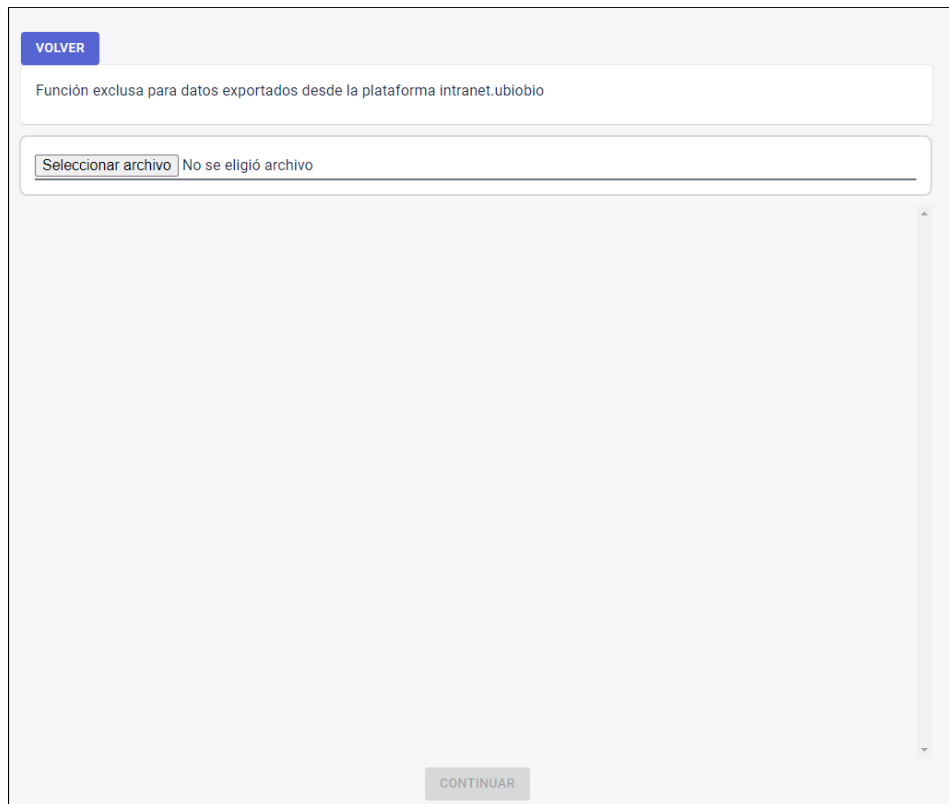
Tras hacer click sobre la cruz en Crear curso, se desplegará un menú como el siguiente el cual permite crear un curso a nombre de un profesor completando el formulario.

The form for creating a course is displayed in a light blue frame. It contains the following elements:

- A blue 'VOLVER' button at the top.
- A label 'Rut Profesor' above an empty text input field.
- A label 'Nombre Asignatura' above an empty text input field.
- A label 'Año' above a text input field containing the value '2021'.
- A grey 'CONTINUAR' button at the bottom.

Figura 13.2.3 Crear Curso Profesor

Tras hacer click sobre la cruz en “Importar Curso” por archivo .xlsx se accede al menú para subir el archivo con la lista de estudiantes (cabe aclarar que el software sólo acepta los archivos exportados desde la plataforma intranet.ubiobio). Una vez seleccionado el archivo, este será leído y mostrará en formato de tarjetas cada dato de los estudiantes a manera de que el profesor pueda revisar que se va a realizar la importación de manera correcta. Al presionar continuar se mostrará el formulario anterior para la creación del curso al cual serán importados estos Alumnos. Finalmente, se mostrará una ventana a manera de confirmación de lo realizado anteriormente junto con el botón para empezar a crear las cuentas, cuadernos e inscripciones de los alumnos en el curso creado en el paso anterior.



The screenshot shows a web interface for importing a course. At the top left is a blue button labeled "VOLVER". Below it is a light blue box containing the text "Función exclusiva para datos exportados desde la plataforma intranet.ubiobio". Underneath is a file selection area with a button labeled "Seleccionar archivo" and the text "No se eligió archivo". At the bottom center is a grey button labeled "CONTINUAR".

Figura 13.2.4 Importar Curso Profesor

VOLVER

Preparando a 27 Alumnos para ser importados...

Rut Profesor

Nombre Asignatura

Año

CONTINUAR

Figura 13.2.5 Crear Curso Importado Profesor

VOLVER

Se han procesado 27 Alumnos. Y están listos para ser agregados al curso "Introducción a la Programación". En el siguiente paso se crearán las cuentas, cuadernos y se inscribirá a los alumnos en el curso. Se podrá acceder al curso posteriormente en el menú de cursos. Considere que esta acción pueda tardar varios minutos

SUBIR

CONTINUAR

Las credenciales para los estudiantes estarán compuestas de su rut para el usuario y la contraseña será el apellido materno seguido del año (ejemplo: Ramirez2021)

Figura 13.2.6 Completar Importación

Tras hacer click sobre el icono de ojo en la tabla curso mostrará todos los datos del curso, junto con estadísticas grupales para poder ver el desempeño del curso y los alumnos que no han participado con la opción de poder eliminar un estudiante y la opción para poder revisar el cuaderno individual del alumno. En este apartado se mostrará de amarillo con un “Problema Pendiente” en caso de que el alumno no haya realizado aún la actividad, “Completado Erróneamente” si el alumno mandó de manera errónea el problema y un “Completado Exitosamente” si lo realizó de manera correcta el problema.

The screenshot shows a web interface for course management. At the top, there are two buttons: 'VOLVER' and 'AGREGAR ESTUDIANTES'. Below them is a section titled 'ESTADÍSTICAS DEL CURSO' with the following data:

- Total de Problemas Resueltos: 2 / 54
- Total de Problemas Correctos: 1 / 2
- Total de Problemas Incorrecto: 1 / 2
- Alumnos Que No Han Participado: 27 [Ver Alumnos](#)

Below the statistics is a table for the course 'Intro 3'. The table has columns for 'Actions', 'Rut', 'Nombre', 'Carrera', and 'Correo'. There are two rows of student data:

Actions	Rut	Nombre	Carrera	Correo
	18987529-0	Gerson Pereira Fontalba	(2927) INGENIERÍA CIVIL EN INFORMÁTICA	gerson.pereira1501@alumnos.ubiobio.cl
		CLAUDIA	(2927) INGENIERÍA CIVIL EN INFORMÁTICA	@ALUMNOS.UBIOBIO.CL

Figura 13.2.7: Participantes de un Curso

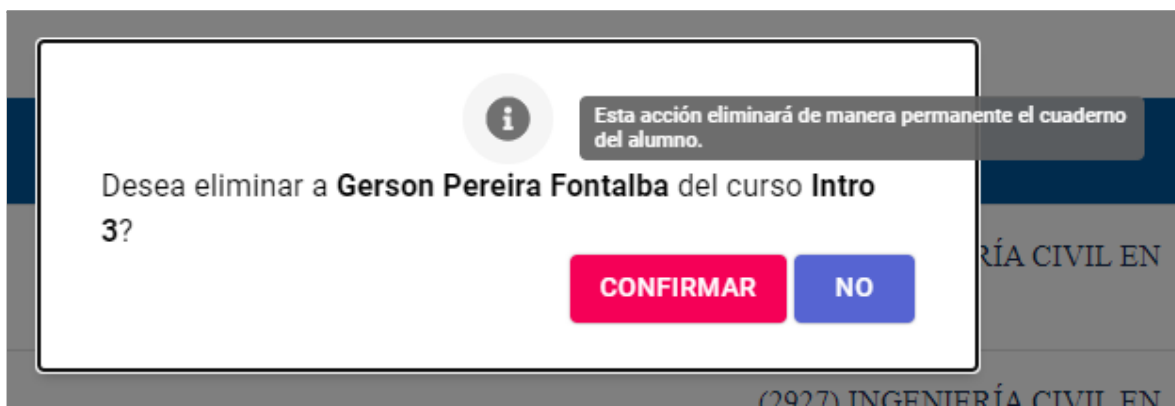


Figura 13.2.8: Eliminar un Alumno del Curso

✔ Ejercicios Correctos: 0
✘ Ejercicios Incorrectos: 0
● Ejercicios Sin Responder: 2

Problema Pendiente

Problema 44: Resolver Mínimo Común divisor entre 2 números ingresados
 Restricción: limitarse al uso de if, else, do-while
 Descripción: mediante 2 datos de entrada calcular el MCD entre ambos

Problema Pendiente

Problema 42: Hola Mundo
 Restricción: sin restricciones
 Descripción: Programar un código que muestre en consola un "Hola Mundo es un programa sin entradas"

CERRAR

Figura 13.2.9: Cuaderno Pendiente Alumno.

Completado Erróneamente

Problema 44: Resolver Mínimo Común divisor entre 2 números ingresados
 Restricción: limitarse al uso de if, else, do-while
 Descripción: mediante 2 datos de entrada calcular el MCD entre ambos

Respuesta:

```

1 #include <stdio.h>
2
3 int main(){
4
5     int a = 0;
6     int b = 0;
7     int c = 0;
8     int d = 0;
9
10    scanf("%d", &a);
11    scanf("%d", &b);
12    scanf("%d", &c);
13    scanf("%d", &d);
14    printf("A=%d B=%d C=%d D=%d", a,b,c,d);
15
16    return 0;
17
18 }
```

Completado Exitosamente

Problema 42: Hola Mundo
 Restricción: sin restricciones

CERRAR

Figura 13.2.10: Cuaderno Resuelto Alumno.

b) Problemas

En la opción del menú Problemas se encontrará la opción para la creación de un problema al apretar la cruz en Agregar Problema junto con la “Lista de Ejercicios” que contempla todos los ejercicios disponibles para agregar a los cuadernos de los estudiantes, permitiendo ver la solución de este al presionar el símbolo con forma de ojo y al presionar la figura con forma de basurero saldrá la opción para confirmar y borrar el problema (esta opción debe ser usada sólo para cuando un problema esté mal subido a la plataforma, ya que lo eliminará de los cuadernos de los estudiantes).

Actions	Autor	Título	Dificultad	Restricción	Variables	Respuesta
	Carlos	Número entre 100-200	3	Uso de las estructuras de selección if-then-else.	150	Está dentro del intervalo
	Carlos	Si un numero es divisor del otro	4	Uso de las estructuras de selección if-then-else.	10 5	Es divisor 10 de 5
	Carlos	Función recursiva que regrese el máximo común divisor de X y de Y	5	Use recursividad para este problema, así como las estructuras de control selectivas y repetitivas.	100 50	El MCD de x=100 y y=50 es: 50
	Carlos	Test Variables de Entrada	2	sin restricciones	1 2 3 4	A=1 B=2 C=3 D=4
	Carlos	Hola Mundo	1	sin restricciones		Hola Mundo es un programa sin entradas

Figura 13.2.11: Menú de Problemas Profesor

```

4 {
5 int Numero1=0, Numero2=0; //Variables
6
7 scanf("%i", &Numero1); //indica sin signo
8 scanf("%i", &Numero2); //indica sin signo
9 if (Numero1 > Numero2)
10 {
11 if (Numero1%Numero2 == 0)
12 {
13 printf("Es divisor %i de %i", Numero1,
14 Numero2);
15 }
16 else
17 printf("%i no es divisor de %i ", Numero2,
18 Numero1);
19 }
20 else
21 printf("No es divisor porque es mayor %i",
22 Numero2);
23 return 0;
24 }

```

CERRAR

Figura 13.2.12: Mostrar Respuesta del Profesor.

Agregar Problema cuenta con un formulario el cual solicita Título, Descripción, Restricción, dificultad y variables de entrada en caso de ser necesarias. Siguiendo de una ventana donde se debe subir el archivo o código que es la solución al problema, se requerirá que el código compile, para lo cual se debe presionar el botón compilar lo cual ejecutará el código subido por el profesor para que el pueda corroborar que la respuesta es la esperada. Finalmente, se podrá ver una ventana donde se verá toda la información y solución del problema para terminar siendo subido a la piscina de problemas para que cualquier profesor pueda hacer uso de la pregunta.

VOLVER

Título del Problema

Descripción del Problema

Restricción

Dificultad del Problema

★★★★★
La dificultad es / 5

Variables de Entrada (No? Dejar vacío)

CONTINUAR

Figura 13.2.13: Agregar Problema Profesor

Una vez agregado los datos en la ventana anterior se visualizará de este modo en esta ventana.

VOLVER

Titulo: Número entre 100 y 200
Descripción: ingresado un número ver si este esta entre 100 y 200
Restricción: Uso de las estructuras de selección if-then-else.
Dificultad: 3 / 5 ★
Variables: 150

SELECCIONAR ARCHIVO:

Seleccionar archivo Problema de Intervalos.c

COMPILAR

Consola:

Está dentro del intervalo

SUBIR PREGUNTAR

Figura 13.2.14: Procesar Pregunta Profesor.

VOLVER

Título
Número entre 100 y 200

Descripción
ingresado un número ver si este esta entre 100 y 200

Restricción
Uso de las estructuras de selección if-then-else.

Variables
150

Dificultad
3 / 5 ★

Resultado
Está dentro del intervalo

Código

```

1 #include <stdio.h>
2
3 #define Limite_Inferior 100
4 #define Limite_Superior 200
5 int main()
6 {
7     float Numero=0; //Definimos nuestra variable
8     //Lee el número ingresado
9     scanf("%f", &Numero);
10    if (Numero >= Limite_Inferior)
11    {
12        if (Numero <= Limite_Superior)
13        {
14            printf("Está dentro del intervalo");
15        }
16    } else

```

CONTINUAR

Figura 13.2.15: Confirmar Problema y Respuesta Profesor.

C) Enviar Problema

En la opción del menú Enviar Problema, encontramos 2 métodos para hacer envío de problemas ya sea de manera individual o aleatorio para darle más variedad a los cuadernos de los alumnos de la asignatura y mas abajo encontraremos un pequeño tip que indica la función de cada modalidad.



Figura 13.2.16: Envío de Problemas Profesor.

Al hacer click en la cruz de la opción enviar problema de forma individual nos aparecerá el siguiente formulario, el cual se rellenará de forma automática al elegir el problema desde seleccionar problema, usted sólo deberá seleccionar el curso al cual desea enviar el problema.

VOLVER

SELECCIONAR PROBLEMA

Titulo del Problema

Descripción del Problema

Restricción

Seleccionar Curso

CONTINUAR

Detailed description: The image shows a vertical form interface for submitting a problem. At the top left is a red button labeled 'VOLVER'. Below it is a blue button labeled 'SELECCIONAR PROBLEMA'. The form contains four input fields: 'Titulo del Problema', 'Descripción del Problema', 'Restricción', and 'Seleccionar Curso'. The 'Seleccionar Curso' field is a dropdown menu with a downward arrow. At the bottom is a grey button labeled 'CONTINUAR'.

Figura 13.2.17: Enviar Problema Individual Profesor

CERRAR

Lista de Ejercicios Q Search X

Actions	Autor	Título	Descripción	Dificultad	Restricción	Variables	Respuesta
+ ○	Carlos	Número entre 100~200	Diseñe un algoritmo que, dado un número real que entra como dato, nos indique si está contenido dentro de los límites predeterminados. El límite inferior es de 100 y el superior de 200.	3	Uso de las estructuras de selección if-then-else.	150	Está dentro del intervalo
+ ○	Carlos	Si un numero es divisor del otro	Programa un algoritmo que, dados dos números enteros que entran como datos, indique si uno es divisor del otro.	4	Uso de las estructuras de selección if-then-else.	10 5	Es divisor 10 de 5

Figura 13.2.18: Seleccionar Problema a Enviar Profesor.

VOLVER

SELECCIONAR PROBLEMA

Título del Problema
Si un numero es divisor del otro

Descripción del Problema
Programa un algoritmo que, dados dos números enteros que entran como datos, indique si uno es divisor del otro.

Restricción
Uso de las estructuras de selección if-then-else.

Seleccionar Curso
POO

CONTINUAR

Figura 13.2.19: Enviar Problema Individual.

Al hacer click en el módulo de enviar problema de forma aleatoria, el profesor debe seleccionar sólo la dificultad del problema que quieren enviar y el curso que quiere que lo reciba, la ventana se vería del siguiente modo.



Formulario de selección de dificultad y curso para enviar un problema de forma aleatoria. El formulario contiene un botón rojo "VOLVER" en la parte superior izquierda. Debajo de este botón, se encuentra el texto "Dificultad Seleccionada" y "Seleccionar Dificultad" con un menú desplegable vacío. A continuación, se encuentra el texto "Seleccionar Curso" con otro menú desplegable vacío. En la parte inferior del formulario, hay un botón gris "CONTINUAR".

Figura 13.2.20: Enviar Problema de Forma Aleatoria Profesor.

d) Agregar Alumno

En la opción del menú Agregar alumno se encuentra la opción para poder ingresar un usuario con rol de alumno y la lista de alumnos con la que cuenta la plataforma para verificar si ya se encuentra ingresado o se ingrese de manera correcta, de manera que el profesor pueda administrar las cuentas de los alumnos que harán uso de la plataforma. Al hacer click en “Ingresar Alumno” se desplegará el formulario para completar los datos de un alumno.



Figura 13.2.21: Menú Alumnos Profesor.

VOLVER

Rut Alumno

77777777-7

Nombre Alumno

Primer Nombre

Apellidos Alumno

Apellido Paterno y Materno

Correo Electrónico

ejemplo@ejemplo.com

Seleccionar Carrera

▼

Contraseña

CONTINUAR

Figura 13.2.22: Agregar Nuevo Alumno.

e) Perfil

En la opción del menú Perfil se encontrará los datos de la cuenta con lo cual se inició sesión de manera que el profesor pueda ver su foto de perfil correspondiente a profesor, rango, correo, título, etc.



Figura 13.2.23: Perfil de Profesor.

13.3 Alumno

Al autenticarse como una sesión de tipo Alumno, se obtendrá acceso a un menú similar al de la siguiente imagen. En este menú se muestra el nombre del usuario con el cual se inició sesión y las opciones que tiene un rol de alumno que consiste en Principal, Mis Cursos y Perfil. Como lo indica el nombre del rol, está creado para poder dar funcionamiento al sistema, ingresar sus respuestas a los ejercicios propuestos por los profesores y poder consultar información de su cuenta y respuestas.



Figura 13.3.1: Menú de Alumno.

a) Principal

Es la 1era pantalla que ve el alumno luego el login en ella se muestran ejemplos y tips de programación en lenguaje C, mediante imágenes de referencia, como se puede apreciar en la imagen.

The dashboard for Gerardo Javier Querales Labarca includes a sidebar with 'Principal', 'Mis Cursos', and 'Perfil'. The main content area is divided into four sections:

- Tipos de variables** (Integer types):
 - Variables de tipo entero

TIPO	BYTES	VALOR MINIMO	VALOR MAXIMO
signed char	1	-128	127
unsigned char	1	0	255
unsigned short	2	-32.768	+32.767
unsigned short	2	0	+65.535
signed int	2	-32.768	+32.767
signed int	2	0	+65.535
signed long	4	-2.147.483.648	+2.147.483.647
unsigned long	4	0	+4.294.967.295

 - Nota: Si se omite el clasificador delante de la variable de tipo entero por defecto se considera "signed"
- Tipos de variables** (Real types):
 - Variables de tipo real

TIPO	BYTES	VALOR MINIMO	VALOR MAXIMO
float	4	3.4E-38	3.4E+38
double	8	1.7E-308	1.7E+308
long double	10	3.4E-4932	3.4E+4932

 - Nota: Las variables de punto flotante son siempre con signo
- Ejemplo 1: Estructura de un programa**
 - Inclusión de bibliotecas, declaraciones de variables y tipos, y secuencia de funciones
 - Una y sólo una función denominada `main` (programa principal)
 - Función: `tipoRetorno Nombre (parametros) (sentencias)`

```

/* ejemplo 1.- Escribe un mensaje en pantalla */

#include <stdio.h> /* incluye biblioteca donde se define E/S */
int main( )
{
    /*Este comentario es ignorado por el compilador y*/
    /*no genera código */
}

```
- Ejemplo 2: Definición de variables**
 - Declaración variable: `tipo Nombre [=valor]`
 - Asignación: `variable = expresión`

```

/* ejemplo 2.- multiplica dos números enteros y muestra el resultado por pantalla */

#include <stdio.h>
int main( )
{
    int multiplicador; /*se define multiplicador como un entero */
    int multiplicando; /*se define multiplicando como un entero */
    int res; /*se define resultado como un entero*/
    multiplicador = 1000; /*se asignan valores*/
}

```

Figura 13.3.2: Dashboard de Alumno.

b) Mis cursos

En la opción del menú de Mis Cursos, aparecerá la información relacionada al curso en el que se encuentra inscrito el alumno, con los datos del profesor que lo dicta y su correo para comunicarse en caso de dudas, también podrá ver un pequeño resumen de los ejercicios que ha realizado, como cuantos se encuentran correctos, incorrectos y pendientes.

The screenshot shows a user interface for 'Mis Cursos'. At the top left is a blue button labeled 'VOLVER'. Below it is a blue header bar with the text 'MIS CURSOS' in white. The main content area is a white box with a light gray border. Inside this box, the course name 'Curso: Intro 3(2021)' is displayed in a large, bold, dark blue font. Below the course name, the professor's name 'Profesor: Brunny Angélica Troncoso Pantoja' and email 'Email: btroncos@ubiobio.cl' are listed in a smaller dark blue font. A horizontal line separates this information from a summary of exercises. The summary consists of three items: a green checkmark followed by 'Ejercicios Correctos: 1', a red circle with a slash followed by 'Ejercicios Incorrectos: 1', and a yellow circle followed by 'Ejercicios Sin Responder: 0'. At the bottom right of the white box is a blue button labeled 'VER CURSO'.

Figura 13.3.3: Mis Cursos de Alumno.

Al hacer click en “ver curso” podrás ver una lista de los problemas asignados en tu cuaderno, cómo cuales has contestado y ver si se encuentran respondidos correctamente.

The screenshot displays a web interface for a student's problem notebook. At the top left is a blue button labeled "VOLVER". Below it is a blue header bar with the text "MIS PROBLEMAS". Underneath is a green header bar with the text "PROBLEMAS RESUELTOS". The main content area lists two solved problems. The first problem is "Problema 44: Resolver Mínimo Común divisor entre 2 números ingresados", with a restriction of "limitarse al uso de if, else, do-while" and a difficulty level of 3 stars (3 yellow, 2 grey). A "SOLUCIÓN" button is located below the problem description. The second problem is "Problema 42: Hola Mundo", with a restriction of "sin restricciones" and a difficulty level of 1 star (1 yellow, 4 grey). A "SOLUCIÓN" button is also present below this problem description.

Figura 13.3.4: Problemas Cuaderno Alumno.

Si tu respuesta es errónea se visualizará de la siguiente manera al dar click en "SOLUCIÓN".

Completado Erróneamente

Descripción: mediante 2 datos de entrada calcular el MCD entre ambos

Tu Respuesta:

```

1 #include <stdio.h>
2
3 int main(){
4
5     int a = 0;
6     int b = 0;
7     int c = 0;
8     int d = 0;
9
10    scanf("%d",&a);
11    scanf("%d",&b);
12    scanf("%d",&c);
13    scanf("%d",&d);
14    printf("A=%d B=%d C=%d D=%d", a,b,c,d);
15
16    return 0;
17
18 }
```

Solución Profesor:

```

1 #include <stdio.h>
2 // Prototipos de la funciones del programa
3 int mcd(int x, int y);
4 int main()
5 {
6     int x,y,t;
7     scanf("%d",&x);
8     scanf("%d",&y);
9     t=mcd(x,y);
10    printf("\n El MCD de x=%d y y=%d es: %d ",x,y,t);

```

CERRAR

Figura 13.3.5: Solución Errónea Alumno.

En caso contrario de ser una respuesta correcta la visualización al dar click en "SOLUCIÓN" se verá de la siguiente manera.

The screenshot displays a web interface for a programming exercise. At the top, a green banner reads "Completado Exitosamente". Below this, the description of the task is shown: "Descripción: Programar un código que muestre en consola un 'Hola Mundo es un programa sin entradas'". The interface is divided into two main sections: "Tu Respuesta:" and "Solución Profesor:". Both sections contain a code editor with the following C code:

```

1 #include <stdio.h>
2
3 int main(){
4
5     printf("Hola Mundo es un programa sin entradas");
6
7     return 0;
8
9 }
    
```

At the bottom of the interface, there is a blue button labeled "CERRAR".

Figura 13.3.6: Solución Correcta Alumno.

Si no has respondido ninguno de los ejercicios anteriores o propuestos en tu cuaderno por el profesor, se visualizará de la siguiente manera, al dar click en “RESOLVER” se podrá acceder al apartado para subir tu código de respuesta.

The screenshot displays a web interface for a student's problem set. At the top left is a blue button labeled 'VOLVER'. Below it is a blue header bar with the text 'MIS PROBLEMAS'. The main content area contains two problem entries, each in a white box with a light gray border. The first entry is 'Problema: 44: Resolver Minimo Comun divisor entre 2 numeros ingresados', with a restriction 'limitarse al uso de if, else, do-while' and a difficulty level of 3 yellow stars and 2 gray stars. A blue 'RESOLVER' button is centered below the problem. The second entry is 'Problema: 42: Hola Mundo', with no restrictions and a difficulty level of 1 yellow star and 4 gray stars. A blue 'RESOLVER' button is also centered below this problem. At the bottom of the interface is a green bar with the text 'PROBLEMAS RESUELTOS'.

Figura 13.3.7: Mis Problemas Alumno.

Al hacer ingreso al apartado de subir respuesta, se verá de la siguiente manera. En seleccionar archivo debes escoger tu código de respuesta, luego debes presionar compilar para saber si tu código ejecuta correctamente lo que pide el profesor, lo cual te indicará con un mensaje en la barra lo que entrega tu código, posteriormente puedes presionar continuar para registrar tu respuesta final, igualmente tienes la opción de cambiar respuesta si la solución ingresada es errónea.

The screenshot shows a web interface for submitting a response to a programming question. At the top left is a blue button labeled "VOLVER". The main heading is "Pregunta 44: Resolver Minimo Comun divisor entre 2 numeros ingresados". Below this, it says "mediante 2 datos de entrada calcular el MCD entre ambos" and "Restricciones: limitarse al uso de if, else, do-while". The difficulty is indicated by five stars, with three yellow and two grey. Under "Adjuntar Respuesta", there is a file selection button labeled "Seleccionar archivo" and a text input field containing "Minimo comun divisor.c". Below the input is a blue button labeled "COMPILAR". Under "Consola:", there is a black bar with green text that reads "El MCD de x=100 y y=50 es: 50". At the bottom center is a red button labeled "CONTINUAR".

Figura 13.3.8: Ingreso Respuesta Alumno.

c) Perfil

En la opción del menú Perfil se encontrará los datos de la cuenta con lo cual se inició sesión de manera que el alumno pueda ver el icono de perfil correspondiente a alumno, nombres, apellidos, correo, carrera, etc.

14 ANEXO: ESPECIFICACIÓN DE LAS PRUEBAS

14.1 Pruebas de Unidad

14.1.1 <Ingreso de Problemas>

Para probar esta unidad solo se necesita la configuración estándar del computador en términos de hardware como de software.

ID caso de prueba	Característica a probar	Datos de entrada					Salida esperada	Salida Obtenida	Éxito/ Fracaso	Observaciones
		título del problema	descripción	restricciones	dificultad problema	variables entrada				
1	Funcionalidad	mínimo común divisor	Cree un código que obtenga mínimo común divisor	use sólo números positivos	2/5	100 50	Problema registrado	Problema registrado	Éxito	Sin observaciones
2	Funcionalidad	mínimo común divisor	Cree un código que obtenga mínimo común divisor			100 50	Botón continuar deshabilitado	Botón continuar deshabilitado	Éxito	Sin observaciones

Tabla 7: Especificación de prueba ingresar problema

14.1.2 <Ingreso de casos de Pruebas>

Para probar esta unidad solo se necesita la configuración estándar del computador en términos de hardware y software.

Precondición:

Debe existir por lo menos un código de respuesta de profesor ingresado.

ID caso de prueba	Característica a probar	Datos de entrada		Salida esperada	Salida Obtenida	Éxito/ Fracaso	Observaciones
		Valores de entrada	valores de salida				
1	Funcionalidad	100 200	100	Datos correctos	Datos correctos	Éxito	Sin observaciones
2	Funcionalidad		100 50	Datos incorrectos	Datos incorrectos	Éxito	Sin observaciones

Tabla 8: Especificación prueba ingreso de casos de prueba

14.1.3 <Ingreso de Respuesta de Alumno>

Para probar esta unidad sólo se necesita la configuración estándar del computador en términos de hardware y software.

Precondición:

- ❖ Debe existir por lo menos un código de respuesta de profesor ingresado para comparar con alumno.
- ❖ Debe haberse creado un cuaderno para el alumno.
- ❖ El cuaderno del alumno debe contener mínimo un problema.

ID caso de prueba	Característica a probar	Datos de entrada		Salida esperada	Salida Obtenida	Éxito/ Fracaso	Observaciones
		Valores de entrada	valores de salida				
1	Funcionalidad	Código de respuesta alumno	Código de alumno procesado	Datos correctos para variables de entrada	Datos correctos para variables de entrada	Éxito	Problema resuelto exitosamente
2	Funcionalidad	Código de respuesta alumno	Código de alumno procesado	Datos correctos para variables de entrada	Datos incorrectos	Fracaso	Código ingresado por alumno no cumple con lo requerido en el problema, problema erróneo

Tabla 9: Especificación prueba ingreso de respuesta alumno

14.1.3 <Ingreso de Datos Alumnos>

Para probar esta unidad solo se necesita la configuración estándar del computador en términos de hardware y software.

Precondición:

- ❖ Debe existir un login activo como profesor o administrador del sistema
- ❖ Debe existir una carrera registrada en el sistema

ID caso de prueba	Característica a probar	Datos de entrada	Salida esperada	Salida Obtenida	Éxito/ Fracaso	Observaciones
		Valores de ingreso				
1	Funcionalidad	Datos de alumnos	Datos ingresados correctamente	Datos ingresados correctamente	Éxito	Alumno registrado en el sistema y su perfil activado
2	Funcionalidad	-Datos de alumnos	Datos correctos para variables de entrada	Datos incorrectos	Fracaso	El alumno no pudo ser registrado

Tabla 10: Especificación prueba ingreso de datos de alumno

14.1.4 <Ingreso de Datos Profesor>

Para probar esta unidad sólo se necesita la configuración estándar del computador en términos de hardware y software.

Precondición:

- ❖ Debe existir un login activo cómo administrador del sistema

ID caso de prueba	Característica a probar	Datos de entrada	Salida esperada	Salida Obtenida	Éxito/ Fracaso	Observaciones
		Valores de ingreso				

1	Funcionalidad	Datos de profesor	Datos ingresados correctamente	Datos ingresados correctamente	Éxito	Profesor registrado en el sistema y su perfil activado
2	Funcionalidad	Datos de profesor	Datos ingresados correctamente	Datos incorrectos	Fracaso	El profesor no pudo ser registrado

Tabla 11: Especificación prueba ingreso de datos de profesor

14.1.5 <Ingreso de Curso>

Para probar esta unidad sólo se necesita la configuración estándar del computador en términos de hardware y software.

Precondición:

- ❖ Debe existir un login activo cómo administrador o profesor del sistema

ID caso de prueba	Característica a probar	Datos de entrada	Salida esperada	Salida Obtenida	Éxito/ Fracaso	Observaciones
		Valores de ingreso				
1	Funcionalidad	Datos del curso(profesor a cargo, nombre, año)	Datos ingresados correctamente	Datos ingresados correctamente	Éxito	curso registrado en el sistema para su uso.
2	Funcionalidad	Datos del curso(profesor a cargo, nombre, año)	Datos ingresados correctamente	Datos incorrectos	Fracaso	El curso no pudo ser agregado al sistema.

Tabla 12: Especificación prueba ingreso de datos de curso

14.1.6 <Ingreso de Carrera>

Para probar esta unidad sólo se necesita la configuración estándar del computador en términos de hardware y software.

Precondición:

- ❖ Debe existir un login activo como administrador

ID caso de prueba	Característica a probar	Datos de entrada	Salida esperada	Salida Obtenida	Éxito/ Fracaso	Observaciones
		Valores de ingreso				
1	Funcionalidad	Datos de carrera(código carrera, nombre carrera)	Datos ingresados correctamente	Datos ingresados correctamente	Éxito	carrera registrada en el sistema para su uso.
2	Funcionalidad	Datos de carrera(código carrera, nombre carrera)	Datos ingresados correctamente	Datos incorrectos	Fracaso	La carrera no pudo ser agregada al sistema.

Tabla 13: Especificación prueba ingreso de datos de carrera.

15 ANEXO: DICCIONARIO DE DATOS DEL MODELO DE DATOS

De acuerdo al modelo de datos desarrollado durante el proyecto, su diccionario de datos es el siguiente:

➤ Alumno : Son los usuarios a los cuales está enfocada el proyecto, ya que son ellos los que acceden a completar y resolver los cuadernos de ejercicios.

Los datos que se registran son los siguientes:

- rut_alumno: Es el identificador único que se le asigna a cada alumno registrado, para reconocerlo dentro el sistema (está compuesta por caracteres).
- nombre_alumno: Corresponde a los nombres del alumno (está compuesto por caracteres).
- apellido_alumno: Corresponde a los apellidos del alumno (está compuesto por caracteres).
- email_alumno: Es el correo institucional asignado al alumno al momento de su inscripción en la universidad (está compuesto por caracteres).
- contraseña_alumno: Corresponde a la contraseña con la que inicia sesión el alumno (está compuesta por caracteres).

➤ Problema: Son los ejercicios que se almacenan en la base de datos para luego ser asignados a cada alumno, sin los problemas no existiría un cuaderno de ejercicios y viceversa.

Los datos que se requieren para registrar un problema son los siguientes.

- **id_problema:** Es un índice que le asigna el sistema a cada problema registrado, por lo que es un valor autoincremental único, por lo que no hay iguales (está compuesto por números).
- **titulo_problema:** Corresponde al título del problema (está compuesto por caracteres).
- **dificultad_problema:** Es una variable que indica la complejidad de los problemas, según su nivel de dificultad desde 1 hasta 5 máximo(está compuesto por números).
- **descripcion_problema:** Corresponde a la descripción del problema donde se indica todo lo que debe hacer el alumno(está compuesto por caracteres).
- **codigo_problema:** Es la respuesta en código C al problema, subida por el profesor y almacenada en la base de datos(está compuesto por caracteres).
- **restriccion_problema:** Corresponde a las restricciones otorgadas al problema por el profesor para aumentar o disminuir su dificultad(está compuesto por caracteres).
- **variables_problema:** Son las variables de entradas definidas con las que se prueba el código de respuesta al compilarlo(está compuesto por caracteres).
- **respuesta_problema:** Corresponde a la respuesta idónea que debería tener el problema una vez compilado, el sistema ejecuta la compilación y guarda esta respuesta de forma automática al registrar el ejercicio(está compuesto por caracteres).

➤ Cuaderno: Se refiere al sitio en donde el alumno puede visualizar cómo responder los problemas asignados para él.

Los datos que se registran de un cuaderno son los siguientes :

- id_cuaderno: Es un índice que le asigna el sistema a cada cuaderno que se crea de manera automática al registrar un alumno(está compuesto por números).
- id_curso: Corresponde al índice del curso en el cual fue registrado el alumno, se registra automáticamente por el sistema (está compuesto por números).
- rut_alumno: Corresponde al rut del alumno, el sistema lo registra automáticamente y es el identificador asignado para registrar y encontrar su cuaderno (está compuesto por caracteres)

➤ Carrera: Se refiere al dato de identificación y nombre de la carrera de la Universidad a la cual pertenece el alumno.

Los datos que se registran en carrera son los siguientes:

- id_carrera: Corresponde al código de la carrera(está compuesto por números).
- nombre_carrera: Corresponde al nombre de la carrera(está compuesto por caracteres).

➤ Curso: Son las asignaturas que se dictan en las diferentes carreras en las cuales se utiliza y evalúa contenido de programación en lenguaje C.

Los datos que se registran en curso son los siguientes.

- id_curso: Corresponde al código del curso(está compuesto por números).

- nombre_curso: Corresponde al nombre de la asignatura(está compuesto por caracteres).
- ano_curso: Corresponde al año en que se registró la asignatura en el sistema, el sistema otorga el año de manera automática(está compuesto por números).

➤ Realiza: Es una relación que contiene atributos propios, registra las respuesta de los alumnos a los distintos problemas que se le asignan.

Los datos que registra realiza son los siguientes:

- codigo_respuesta: Es el código de respuesta en lenguaje C el alumno subido a la plataforma(está compuesto por caracteres).
- respuesta_realiza: Corresponde a la respuesta obtenida luego de que el sistema compila y ejecuta el código del alumno(está compuesto por caracteres).
- calificacion: Corresponde a un dato de tipo tinyint, que asigna si la respuesta del alumno es correcta o no, mediante un sistema booleano de 1 correcto y 0 incorrecto(está compuesto por números).

➤ Profesor: Son las personas que pueden acceder al sistema para ingresar nuevos problemas y soluciones, cursos y alumnos, además de poder obtener datos de rendimientos del curso(el administrador también se encuentra inscrito en esta tabla, pero asignado con datos diferentes en algunos campos).

Los datos que se registran de profesor son los siguientes:

- rut_profesor: Es el identificador único que se le asigna a cada profesor registrado, para reconocerlo dentro el sistema (está compuesta por caracteres).

- nombre_profesor: Corresponde a los nombres del profesor (está compuesto por caracteres).
- apellido_profesor: Corresponde a los apellidos del profesor (está compuesto por caracteres).
- email_profesor: Es el correo institucional asignado al profesor al momento de ser contratado en la universidad (está compuesto por caracteres).
- titulo_profesor: Corresponde al nivel de estudios con los que cuenta el profesor tales cómo ingeniero/a, magister, doctorado, etc(está compuesto por caracteres).
- contraseña_profesor: Corresponde a la contraseña con la que inicia sesión el profesor (está compuesta por caracteres).